mi computer

16

CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

150 ptas.

Editorial Delta, S.A.



MI COMPUTER CURSO PRACTICO
DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona, y comercializado en exclusiva por Distribuidora Olimpia, S.A., Barcelona

Volumen II - Fascículo 16

Director: José Mas Godayol
Director editorial: Gerardo Romero
Jefe de redacción: Pablo Parra
Coordinación editorial: Jaime Mardones
Asesor técnico: Jesús Nebra

Redactores y colaboradores: G. Jefferson, R. Ford, S. Tarditti, A. Cuevas
Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.°, Barcelona-8
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador,* se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-85822-90-0 (tomo 2)
84-85822-82-X (obra completa)
Depósito Legal: B. 52/1984

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 028405
Impreso en España - Printed in Spain - Abril 1984

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, Madrid-34.
Distribuye para Argentina: Viscontea Distribuidora, S.C.A., La Rioja 1134/56, Buenos Aires.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio Blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Ferrenquín a Cruz de Candelaria, 178, Caracas, y todas sus sucursales en el interior del país.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)

Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 16 690 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 087 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 3371872 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Distribuidora Olimpia (Paseo de Gracia, 88, 5.°, Barcelona-8), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Distribuidora Olimpia, en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.



Comunicación por cable

La televisión por cable abre un nuevo camino para que los usuarios de ordenadores personales se comuniquen entre sí



Ian McKinnell

Ciudad para el futuro
Milton Keynes, una ciudad que se construyó empezando desde cero, dispuso desde el principio de televisión por cable. Los 22 000 hogares de la ciudad tienen acceso a siete canales de televisión (seis de programas variados y en directo y uno que sólo emite películas) y a seis canales de radio VHF (Very High Frequency: frecuencia muy alta). El siguiente paso hacia una red de información integrada prevé que los suministros de gas y electricidad se midan centralmente y no en cada edificio. Pronto se agregarán redes de área local que se instalarán en los edificios públicos, así como un sistema comunitario de videotex

Se habla mucho acerca de los beneficios que nos reportaría la denominada "revolución del cable", que nos permitiría contemplar en nuestro hogar 20 o 30 canales de televisión, pero se habla muy poco de un efecto colateral que podría cambiar la forma en que nos comunicamos con los otros miembros de la comunidad. Porque uno de esos canales se podría reservar para que sirviera como vínculo entre los ordenadores personales. Ya hemos analizado dos procedimientos para utilizar éstos como terminales de comunicaciones: los sistemas de videotex (véase p. 268), que permiten acceder a una gran base de datos general, y las redes de área local (véase p. 218).

Utilizando un sistema de videotex, como el Mailbox de Prestel, en Gran Bretaña, les es posible a los suscriptores enviarse mensajes, y cualquiera de ellos podría contratar bienes y servicios (p. ej., vacaciones) directamente del proveedor de información. Sin embargo, estos dos servicios son de aplicación limitada. Del mismo modo, si el ordenador que emplea el usuario forma parte de uno de los centros de trabajo de una red de área local, le será posible comunicarse con cualquier otra estación; pero incluso la más grande y poderosa de las redes basadas en microordenadores abarcará solamente dos o tres kilómetros y es poco probable que tanto el banco como el supermercado y la farmacia de los que el usuario es cliente se hallen dentro de ese perímetro. Por supuesto, éste puede estar suscrito a un sistema de videotex y ser miembro de una red de área local. No obstante, la situación distaría de parecerse a la que disfrutaría si fuera miembro de una comunidad en la que cada hogar se hallara conectado electrónicamente a los demás.

El factor físico más importante que impide la creación de redes de este tipo que cubran ciudades enteras, es la pérdida de señal en el medio de transmisión, que se produce ya sea utilizando pares de cables enroscados, cables coaxiales (como el de la antena de su televisor) o fibras ópticas. Es este fenómeno lo que en la actualidad limita las dimensiones de las redes de área local, y la única forma de superar el problema consiste en insertar estaciones reamplificadoras o "reforzadoras" en la red a intervalos frecuentes.

Una segunda consideración es la de la "densidad del tráfico" o el volumen de información a comunicar. Éste influye en la anchura de banda o alcance de las frecuencias de transmisión requeridas. Por regla general, se necesita una anchura de banda de dos hertzios para cada bit por segundo que se desee transmitir. Una transmisión de 300 baudios (300 bits por segundo) exige 600 Hz; por su parte, una transmisión de 1 200 baudios necesita 2,4 kHz. El habla normal utiliza un mínimo de 3 kHz a través de una línea telefónica. No obstante, la transmisión de una imagen de televisión en color requiere 8

MHz (tres mil veces la del habla). En otras palabras, la anchura de banda que se necesita para transportar una imagen de televisión podría difundir 3 000 conversaciones telefónicas separadas.

La capacidad de un medio de transmisión tal vez se exprese mejor en términos del número de conversaciones telefónicas que puede retener. En Gran Bretaña, el cable de mayor capacidad que emplea actualmente la British Telecom puede transmitir 20 000 conversaciones a la vez, pero en pruebas experimentales dicha empresa ha conseguido evaluar en 500 MHz el cable coaxial, lo que implica que potencialmente es capaz de transportar 167 000 conversaciones simultáneamente. El cable tiene 1 cm de espesor. Se le podría comparar con la tecnología de las fibras ópticas, según la cual una simple hebra de vidrio más delgada que un cabello tiene capacidad para difundir hasta 10 000 conversaciones telefónicas.

El hardware para comunicaciones ya está en el mercado, y crear una red para una población o una ciudad sería una cuestión sencilla. Consideremos ahora cómo podríamos establecer una red informática comunitaria basada en el microordenador BBC Modelo B utilizando el Econet de Acorn. Cuando analizamos las redes de área local, señalamos que el controlador de la red no podía estar situado a más de 500 metros de ningún centro de trabajo, lo que implica que nuestra red puede cubrir una circunferencia de un kilómetro de diámetro. También vimos que puede dar cabida hasta a 254 miembros, dejando una estación dedicada a servir el archivo (manejando el disco comunitario) y otra para controlar la impresora. Pero si reserváramos otra estación como canal de comunicación con una segunda red, entonces un programa relativamente sencillo nos permitiría conectar dos redes entre sí. La conexión requeriría dos máquinas dedicadas a pasar mensajes de una red a la otra, comunicándose a través de sus respectivas conexiones en paralelo. Cuanto mayor fuera el número de máquinas que estuviéramos dispuestos a reservar con este fin, más redes podríamos conectar entre sí.



Cortesía de MTV

Estación satélite
MTV utiliza un sistema de transmisión por satélite para emitir programas a sus 1 650 compañías afiliadas, que luego los envían por cable a los hogares de los trece millones y medio de suscriptores. MTV proporciona 168 horas de programación semanal, toda en estéreo. Se incluyen 8 minutos de publicidad por hora de emisión. Aunque el servicio es estrictamente en una sola dirección, la empresa fomenta un cierto grado de interacción de la audiencia a través de números de teléfono gratuitos. Con la introducción del tráfico por cable en dos direcciones, este tipo de acción recíproca instantánea de los televidentes se generalizará mucho más

Por supuesto, ésta es una solución muy provisional al problema de enlazar por cable una comunidad entera, pero si la idea tuviera verdadera aceptación se podría reemplazar por una conexión de red construida especialmente. Sin embargo, es poco probable que se pudiera realizar una conexión de este orden dado que requiere que todos los miembros utilicen el mismo tipo de microordenador. Para que sea realmente eficaz, una red así ha de ser completamente "transparente", lo que significa que debe permitir que un Spectrum, por ejemplo, se comunique con un Dragon 32. Ello exige un controlador central del sistema que manipule la conversión entre los diversos protocolos que utilizan los distintos fabricantes de ordenadores. Asimismo, el controlador podría ser el punto de enlace con el Prestel y otros servicios de videotex, el sistema bancario y las entidades financieras, el servicio sanitario y otros servicios públicos, y todas las otras áreas de la sociedad que se hubieran informatizado sin contar antes con la compatibilidad entre las máquinas.

¿Quién proporcionaría los fondos para instalar y hacer funcionar un servicio de tales características? La experiencia británica sugiere que éste es el punto más conflictivo. Cuando el gobierno británico formuló propuestas por primera vez para la instalación de un sistema de televisión por cable, puso especial énfasis en los beneficios que tal sistema reportaría a la industria de la tecnología de la información, y, en consecuencia, varios usuarios potenciales de dicho sistema se abocaron a proyectos de investigación. Una cadena nacional de supermercados, en un gesto de audacia, puso en marcha un sistema piloto propio de telecompra para probar su viabilidad y la reacción del público ante el mismo. El esquema no sobrevivió ni siquiera a su breve período de prueba, y en todas partes se llegó a la misma conclusión: los servicios de esas características tenían poca aceptación.

Cuando se presentó en el Parlamento británico el borrador del proyecto de ley proponiendo la instalación de la televisión por cable, aún contenía referencias a servicios informáticos como "telecompra" y "telebanco", pero decía más abiertamente que la dirección del sistema del cable sólo habría de "fomentar la provisión de servicios informáticos en dos direcciones". En consecuencia, se concedía mayor importancia a la tecnología del entretenimiento que a la de la información. De hecho, a los operadores del cable se les prohibiría expresamente ofrecer cualquier forma de servicio telefónico, ni siquiera facilidades para videoconferencias, si bien el borra-

FIBRA ÓPTICA

CABLE COAXIAL

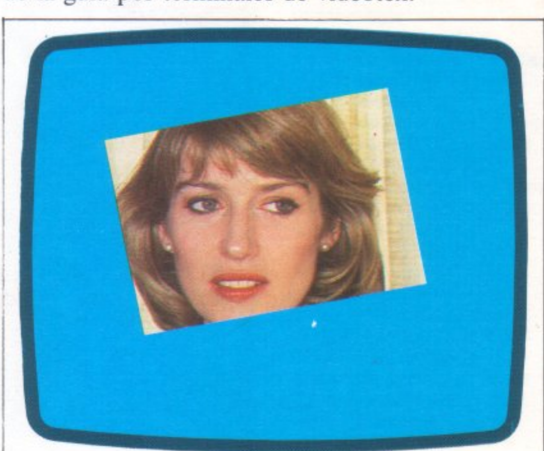
Gary Marsh

Impulso rápido
Las fibras ópticas, que se desarrollaron por primera vez en 1966 en los Standard Telecommunication Laboratories de Harlow (Inglaterra), se basan en el fenómeno de la reflexión interna total. La luz introducida por un extremo viajará a través de la fibra de vidrio con una pérdida muy escasa de luminosidad, volviéndose a reflejar cada vez que golpea contra la "pared" exterior, al igual que lo haría en un prisma o en una corriente de agua. Para conseguir este resultado, el nivel de pureza del vidrio empleado ha de acercarse al 100 %. Se utiliza la luz de un láser infrarrojo, porque la señal se debe impulsar (encenderse o apagarse) muy rápidamente. Los cables coaxiales, familiares por su utilización para conexiones a la antena de los televisores, constan de un único cabo de alambre de cobre grueso, rodeado por una pantalla de cable más delgado tejido en un tubo. Estos dos conductores están aislados el uno del otro

dor de un nuevo documento contempla la posibilidad de que finalmente la red por cable asuma todos los servicios de telecomunicaciones.

La experiencia de otros países es similar. En Estados Unidos, donde la televisión por cable existe desde hace mucho tiempo, aún se utiliza muy poco el sistema de cable para proporcionar un medio de comunicaciones digital en dos direcciones. Incluso allí donde se emplea parece que se hace más bien con fines triviales.

En los países escandinavos y en Holanda ha habido un movimiento tendente a crear una interconexión de ordenadores personales a nivel comunitario. En la actualidad se encuentran en funcionamiento redes locales de correo electrónico y de contratación de niñeras por medio de ordenador; algunas comunidades incluso efectúan plebiscitos informales acerca de temas de interés local. Lamentablemente, estos beneficios sociales tan evidentes no siempre son tenidos en cuenta, en algunos países, por los patrocinadores de sistemas de red por cable. Tal vez quienes perciban con mayor lucidez los beneficios de las redes de comunicaciones sean aquellos miembros de la comunidad que tienen muy poco interés en adquirir siquiera un sencillo microordenador. Una vez instalados los cables interactivos, se podrían crear incentivos para que las industrias de servicios, como los bancos o el servicio de telecomunicaciones, distribuyan terminales a un costo reducido, o incluso completamente libres de todo cargo, como sucede en algunas regiones de Francia, donde la red nacional de teléfonos ha sustituido las guías telefónicas y el servicio de consulta de la guía por terminales de videotex.



© BBC Stills

¿Vale una imagen más que mil palabras?
Es difícil determinar el número de bits digitales que se necesitan para componer una imagen de televisión en color, pero considerando que la anchura de banda de 9 MHz permite la transmisión de 4,5 millones de baudios, y que cada imagen completa se transmite 25 veces por segundo, el simple cálculo aritmético nos da una cifra de 180 000 bits por fotograma. Por otra parte, la emisión de mil palabras requiere unos 60 000 bits

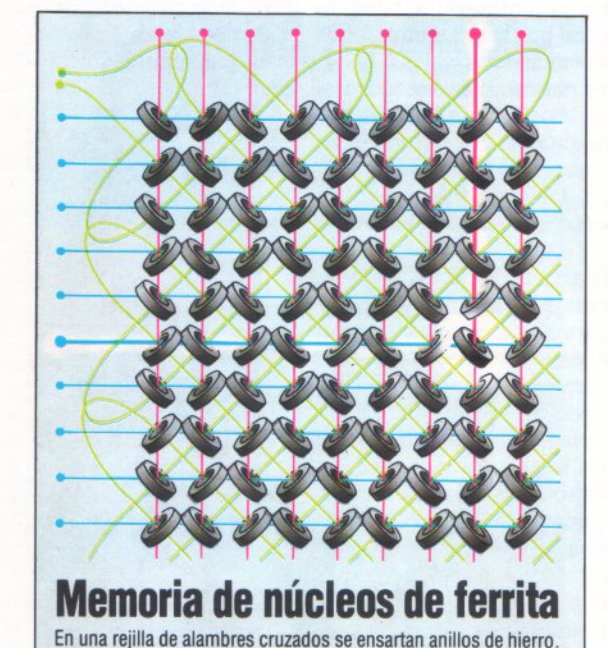
Las nuevas memorias

Hallar medios más compactos de almacenar información ha sido la meta de los diseñadores desde que se inventara el ordenador

Desde que se investigara por primera vez sobre la electricidad, los científicos han estado buscando formas de poder utilizarla para almacenar información. La electricidad se puede concebir o como un flujo de electrones o como una onda en movimiento, pero en ambas descripciones la característica implícita es el movimiento. La imposibilidad de retener algo que por su propia naturaleza siempre ha de estar desplazándose, ha llevado a la adopción de métodos de almacenamiento indirecto. Se han hallado muchas soluciones al problema de manipular datos en forma de señales eléctricas.

Durante la segunda guerra mundial se trabajó intensamente en el radar, para separar la señal reflejada desde un aeroplano en movimiento de los impulsos que reverberaban en objetos fijos, como los árboles. Se inventó un dispositivo, denominado *línea de demora de mercurio*, que podía almacenar temporalmente los impulsos de las ondas de radio y comparar así las ondas reflejadas en los sucesivos "barridos" del radar, de modo que se podían eliminar los patrones permanentes.

Una línea de demora de mercurio se compone de un tubo de vidrio de un metro de longitud, lleno de mercurio, y con un cristal de cuarzo a cada extremo. Cuando se aplica una señal eléctrica en uno de los extremos, el cristal de cuarzo vibra y crea una onda física que viaja hasta el otro extremo del tubo, donde el otro cristal la detecta y la vuelve a convertir en una señal eléctrica. La onda tarda aproximadamente 660 microsegundos en desplazarse a través del tubo, pero volviendo a alimentar la señal en un bucle a través del mercurio, se pueden almacenar los impulsos durante varios minutos antes de que la señal se distorsione demasiado.



Memoria de núcleos de ferrita

En una rejilla de alambres cruzados se ensartan anillos de hierro. Cualquier anillo se puede direccionar enviando una corriente a través de los alambres horizontales y verticales. El alambre lector, que traspasa todos los anillos, recoge los cambios que se producen en el flujo magnético del anillo, indicando si hay almacenado un 1 o un 0

Kevin Jones

Con un reloj que regulaba la secuencia de impulsos a 500 000 ciclos por segundo, se podían almacenar 330 bits en un tubo de un metro. El tamaño de la memoria se podía ampliar alargando el tubo, pero sólo al precio de aumentar el tiempo de acceso, dado que un impulso sólo se podía leer cuando estaba en el circuito eléctrico fuera del tubo. Estos sistemas eran caros y voluminosos, ya que requerían una ingeniería de gran precisión para construir juegos de tubos de una exactitud de una centésima de milímetro.

En Gran Bretaña, F. C. Williams inventó en la Universidad de Manchester otro método para almacenar datos; Williams investigaba el empleo de la electricidad estática generada en la superficie interior de un tubo de rayos catódicos (una pantalla de televisión) como medio de almacenamiento. Un "disparador de electrones" componía en la pantalla un patrón de cargas estáticas, y este patrón se podía detectar por una rejilla de alambre próxima a la superficie exterior de la pantalla de vidrio. Hacia 1947 se podían almacenar 2 048 bits de información durante varias horas en una única pantalla. Aunque la velocidad de acceso era alta, las cargas estáticas se habían de refrescar cada 30 microsegundos, ya que de lo contrario se extinguían y se perdían.

La utilización de la cinta magnética se probó por primera vez con el LEO (véase p. 320) en Gran Bretaña, y en Estados Unidos con el UNIVAC (Universal Automatic Computer) a fines de los años cuarenta. Ésta fue la primera técnica en virtud de la cual se pudo almacenar gran cantidad de información de forma barata y fiable. Se almacenaron ocho bits en un "cuadro" a través de la cinta a una densidad de 2 360 "cuadros" por centímetro. Con una longitud total de cinta de 61 metros o más, se consiguieron memorias permanentes de al menos un megabyte. No obstante, aun con la rapidez de los motores activadores, el tiempo de acceso a los datos en mitad de la cinta era de algunos segundos. En consecuencia, esta clase de almacenamiento halló su utilización natural en aquellos archivos donde los datos se requieren secuencialmente, como en el cálculo de la nómina de una empresa. La cinta se podía acelerar haciendo flotar la "cabeza" que detecta las señales magnéticas en un "colchón" de aire, que también alivia el desgaste por uso y tensión.

En 1950 se inventó en el Massachusetts Institute of Technology una memoria de gran capacidad, gran velocidad de acceso y bajo costo por bit. La memoria "de núcleos de ferrita" empleaba anillos de hierro ensartados en una rejilla de alambres que se intersectaban entre sí. Haciendo pasar una corriente por un alambre que atravesaba un anillo de hierro, el metal se magnetizaba. En el campo mag-

nético del anillo hay dos direcciones posibles y éstas se utilizan para representar los dos estados binarios, 0 y 1. La dirección del magnetismo del anillo se puede "invertir" mediante una corriente adecuada. Y así como una corriente induce magnetismo, también sucede lo contrario: cuando "se invierte" la dirección del magnetismo, se induce una corriente inversa pequeña pero detectable. El campo magnético sólo se invertirá cuando se emplee una corriente por encima de un umbral determinado. Proporcionando la mitad de la corriente umbral a través de un alambre vertical y la otra mitad a través de un alambre horizontal, sólo uno de los anillos del enrejado recibirá suficiente corriente como para invertirse.

El diámetro de los anillos individuales disminuyó con el tiempo de aproximadamente cuatro milímetros a algunas centésimas de milímetro, reduciendo, de este modo, la energía necesaria para escribir datos en una celda. La información de una celda de memoria determinada se leía intentando invertirla. Mediante el control de una corriente inducida se podía descubrir si la celda se había invertido o no y, en consecuencia, se podía deducir su estado original. Sin embargo, este método de lectura colocaba todas las celdas en un estado magnético, de modo que después de cada lectura se debían volver a escribir los datos.

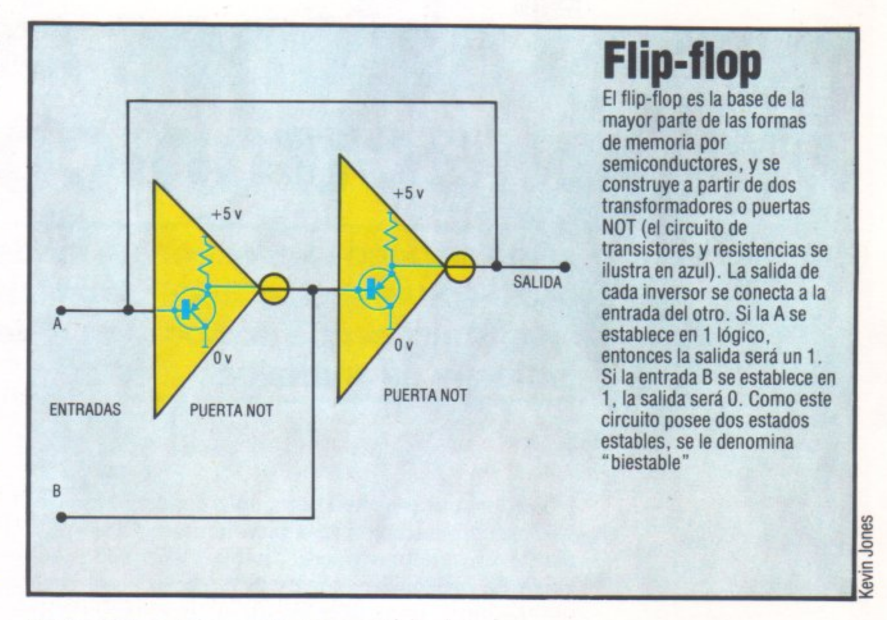
Con la integración a gran escala (LSI) de los transistores en un solo chip, se ha vuelto a restablecer uno de los primeros métodos para retener datos (que se venía utilizando desde el ENIAC): el *flip-flop* (circuito biestable). Un flip-flop es un dispositivo que es estable en uno de dos estados (de allí su característica de "biestable"). Se construye a partir de dos interruptores electrónicos uno detrás del otro, alimentando la salida de cada uno en la entrada del otro. De este modo se puede "atrapar" un impulso en el flip-flop hasta que se lo requiera.

Cada bit necesita dos dispositivos interruptores; además, los primeros flip-flops de válvula eran caros, nada fiables y se quemaban muy a menudo. Sin embargo, con los circuitos integrados, en los cuales en un único chip se incorporan cientos de miles de interruptores transistorizados, el flip-flop ha vuelto a recuperar su importancia.

Los chips de RAM "estática", como los que poseían todos los primeros ordenadores personales, contienen miles de circuitos biestables, uno para cada bit. No obstante, la mayoría de las máquinas más nuevas incorporan una RAM "dinámica". El usuario no nota ninguna diferencia, pero la RAM dinámica es más rápida y su construcción es más barata. En realidad es como una matriz de condensadores, cada uno de los cuales puede almacenar una carga eléctrica. El inconveniente está en que, al tratarse de una carga almacenada, tiende a perderse; por este motivo, los contenidos se han de "refrescar" miles de veces por segundo, lo que se lleva a cabo mediante un sistema de circuitos especial incorporado en el chip.

Los tipos de memoria más recientes que existen en el mercado son la memoria de "burbuja" y la memoria de láser óptico. Esta última es similar a los discos que se utilizan para almacenar música o películas de video y su lectura se realiza por medio de un haz de rayos láser. En la memoria de burbuja, en el interior de un chip de material magnético se crean diminutos campos magnéticos o burbujas. Estas burbujas se pueden mover en bucles y la presencia o la ausencia de una burbuja representa un 1 o un 0 binarios. Los bucles se mantienen pequeños para conseguir un tiempo de acceso rápido. Su densidad es potencialmente tan elevada que este tipo de memoria puede llegar a reemplazar los actuales sistemas de discos y cassettes de superficie móvil.

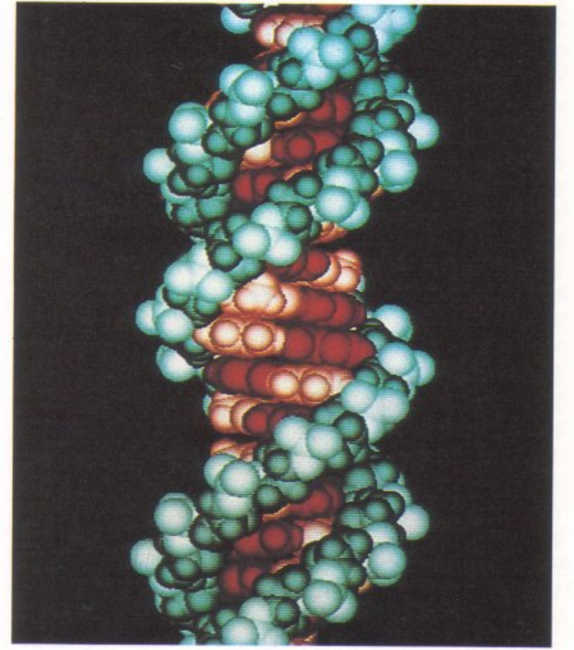
Las memorias criogénicas podrían, asimismo, llegar a ser realidad. Al alcanzar temperaturas próximas al cero absoluto, la resistencia eléctrica en un medio desaparece para proporcionar lo que se denomina superconductividad, y entonces resulta teóricamente posible almacenar una carga equivalente a la de un electrón. Las cargas sobre las que operan las clases actuales de memoria de chips son muy pequeñas, pero aun así cada una de ellas equivale a algunos millones de electrones. Con la superconductividad que se obtiene enfriando estos medios de memoria a temperaturas criogénicas en baños de helio líquido, la velocidad y la capacidad de las memorias alcanzan un punto que supera muchísimo al que ofrecen los sistemas actuales.



Flip-flop

El flip-flop es la base de la mayor parte de las formas de memoria por semiconductores, y se construye a partir de dos transformadores o puertas NOT (el circuito de transistores y resistencias se ilustra en azul). La salida de cada inversor se conecta a la entrada del otro. Si la A se establece en 1 lógico, entonces la salida será un 1. Si la entrada B se establece en 1, la salida será 0. Como este circuito posee dos estados estables, se le denomina "biestable"

Kevin Jones



Biotecnología
Pese a haber alcanzado la electrónica un increíble nivel de miniaturización, toda operación en una memoria por semiconductores implica el flujo de muchos miles de electrones. Los científicos ya están intentando desarrollar memorias que funcionen a nivel molecular, es decir, con electrones individuales, lo cual tiene más relación con el campo de la biotecnología que con el de la microelectrónica. Se sabe, por ejemplo, que la estructura de una molécula de DNA (ácido desoxirribonucleico) almacena el código genético de los seres vivos. No obstante, pasarán muchos años antes de que el hombre sea capaz de reproducir una molécula similar

Tablones de anuncios electrónicos

¿Qué se puede hacer con un modem? Cualquier usuario de ordenador puede acceder a los tablones de comunicaciones electrónicos, que no son sino la versión computerizada de los tablones de anuncios

Un ordenador personal por sí solo tiene una capacidad de comunicación más bien limitada, pero mediante un teléfono puede "hablar" con toda una gama de ordenadores diversos, ya sean poderosos ordenadores universitarios de unidad principal, bases de datos públicas a gran escala o, simplemente, con el ordenador personal de la casa de al lado.

En Gran Bretaña, a través del BBS (Bulletin Board Services: servicio de tablones de comunicaciones), cualquier usuario de ordenador puede hacerles llegar mensajes a otros usuarios, anunciar artículos que desee vender, e incluso intercambiar programas. Un BBS es, en realidad, el equivalente electrónico de los tablones de anuncios que se encuentran en la mayoría de las sedes de clubs o asociaciones. Por lo general están a cargo de voluntarios que utilizan un ordenador personal corriente con un gran dispositivo de almacenamiento (p. ej., un disco), al que puede acceder cualquier persona. La distancia existente entre un ordenador y otro no cuenta: se puede acceder a máquinas de la misma ciudad, de centros urbanos situados en el otro extremo del país e incluso de otro continente. La única restricción, por supuesto, la constituirá el importe de la factura telefónica.

Sin embargo, antes de que el ordenador personal pueda empezar a "hablar" por teléfono, se necesita un elemento de hardware que los conecte a los dos. Dicho dispositivo se denomina *modem* (véase p. 108) y se enchufa en la conexión en serie (RS232), situada en la parte posterior del ordenador. Algunas máquinas, como el Vic-20 y el Spectrum, no poseen dicha conexión; en tal caso, el usuario también tendría que adquirir una ficha de interface en serie para enchufar en la parte posterior. El modem se conecta a la línea telefónica ya sea directamente, a través de un enchufe telefónico extra, o bien acústicamente, empleando un acoplador acústico. Existen dos "velocidades" de modem (velocidades baudio) de uso generalizado: 300 y 1200/75 baudios. Los operadores comerciales, como el Prestel y la British Telecom, utilizan la velocidad mayor, de 1200/75, mientras que el BBS (que es básicamente para ordenadores personales) emplea 300 baudios. Velocidad de 1200/75 significa que la información le llega al usuario a 1200 baudios y que éste, a su vez, la envía al ordenador central a 75 baudios. Por supuesto, es preferible comprar un modem que tenga velocidades baudio conmutables. Si el modem es del tipo de "conexión directa", el usuario también tendrá que utilizar un enchufe extra para él, que le alquilará la British Telecom.

Para hacer funcionar el modem también es necesario obtener software que le permita a la máquina cumplir el cometido de un terminal de ordenador. Existen dos tipos de software de terminal: el "mudo" y el "vivo". Un terminal vivo puede cargar programas y guardar mensajes de otros ordenadores. Un terminal mudo no puede realizar estas funciones, pero es más sencillo y, por lo tanto, más apropiado para ser empleado en los primeros intentos de comunicación por parte del usuario. Este tipo de software existe para los ordenadores personales "más antiguos", como el Apple II y el Tandy TRS80, así como para algunos de los modelos más nuevos; el BTERM es el programa para el BBC Modelo B. En cuanto a otros ordenadores, este software lo suelen proporcionar gratuitamente los grupos de usuarios que los utilizan. En caso de que no existiera, el usuario puede escribir sin excesiva dificultad su propio software para el terminal. Lo que necesita es un programa que envíe todos los caracteres digitados en el teclado a la conexión RS232 y que visualice en la pantalla los datos recibidos desde esta conexión.

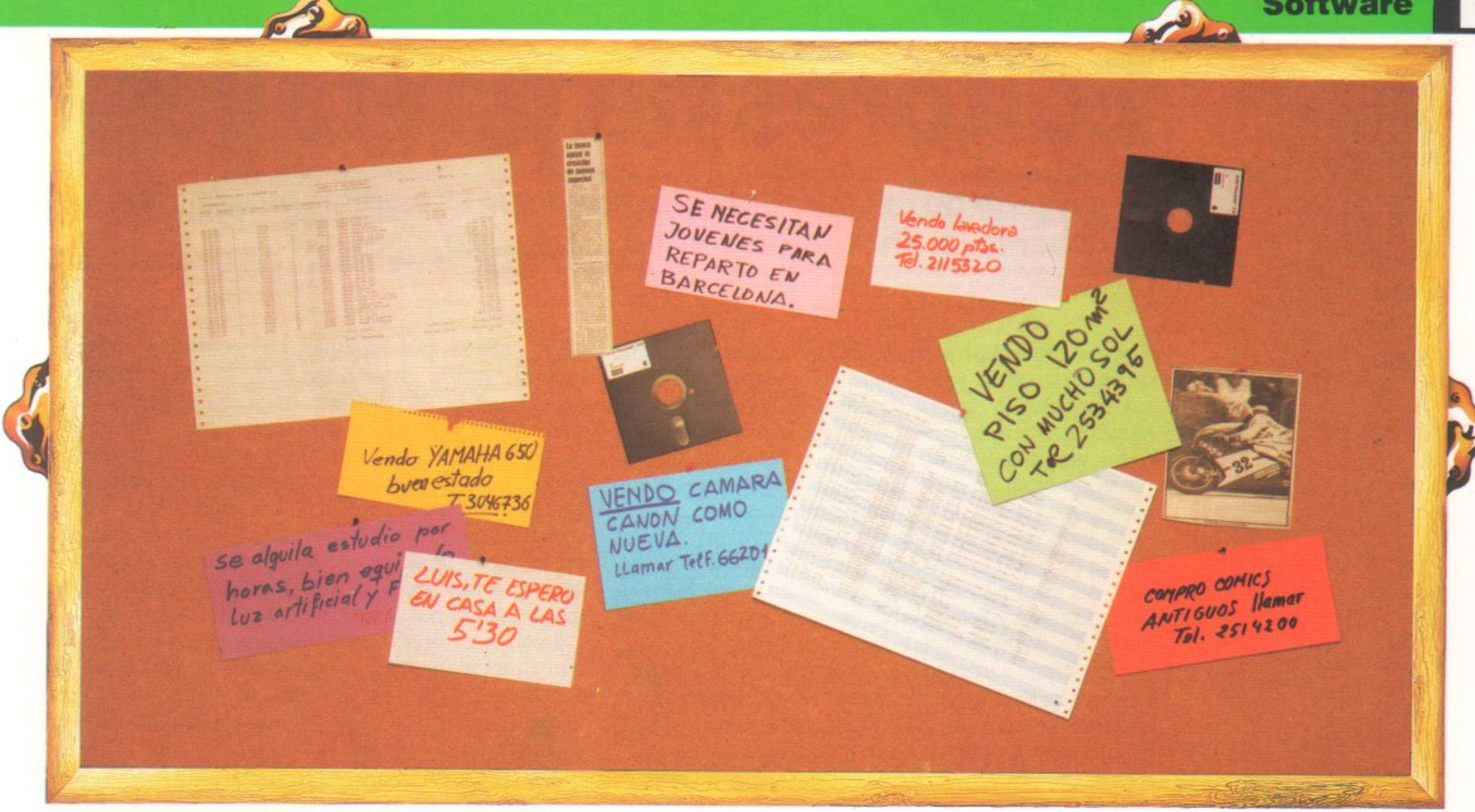
Para ilustrar cómo se utiliza un modem, sigamos los pasos necesarios para comunicarse con un BBS. Primero, el usuario ha de averiguar el número de teléfono y los detalles técnicos (como la velocidad baudio) del ordenador al cual desea acceder. Después de ejecutar el software de comunicaciones en su ordenador, se procede a marcar el número de teléfono. Si el ordenador con el que se desea comunicar está "escuchando", el usuario oirá un tono alto a través del auricular. Éste es el "tono portador". Si el usuario estuviera utilizando un acoplador acústico, únicamente habría que insertar el receptor en las tazas de goma. Con un modem de los de tipo de "conexión directa", usted con sólo pulsar el interruptor marcado "LINE" o "DATA" reemplaza el receptor. En ambos casos, el ordenador debería estar conectado.

Probablemente lo primero que se vea en la pantalla sea algún tipo de "saludo", generado por el ordenador al otro extremo de la línea telefónica. Luego, al "huésped" se le pedirá su identificación y/o contraseña. Si el servicio al que se ha llamado es privado y el usuario no se encuentra registrado en él, entonces probablemente no conseguirá mucho más; ¡a menos que sea un buen adivino o una persona muy paciente! No obstante, muchos ordenadores permiten el acceso restringido a los "huéspedes" que no sean usuarios registrados, de modo que vale la pena probar con algunas palabras como

TELECOM GOLD

Correo electrónico
Telecom Gold es el sistema de correo electrónico de la British Telecom, una versión más sofisticada de los Bulletin Board Services destinada al mercado comercial. Los usuarios inscritos pueden alquilar un "buzón" en un miniordenador de la British Telecom, al que los otros usuarios pueden enviar sus mensajes. Éstos se pueden leer o copiar en el ordenador del usuario. El Telecom Gold permite acceder a otras redes electrónicas en otros lugares del mundo, con la ventaja de que todas las diferencias técnicas existentes entre las máquinas de los usuarios son obviadas automáticamente por los ordenadores que integran la red

Cortesía de British Telecom



NEWUSER (nuevo usuario), GUEST (huésped) o HELP (socorro).

El BBS, por otra parte, está abierto a cualquiera y es gratuito (excepto en lo que atañe al costo de la llamada telefónica). Basta con que el usuario dé su nombre y el de su ciudad cuando se le solicite, y una vez "alistado" se le preguntarán detalles de su ordenador, como el ancho de la pantalla o quizá la marca. Así, el ordenador "anfitrión" lo identificará en futuras llamadas y adaptará el sistema para que funcione adecuadamente con su ordenador.

Una vez cumplido todo ello, se le proporcionará alguna información relativa al sistema, como los horarios de funcionamiento y detalles técnicos, y un límite de tiempo para su llamada (tal vez 30 minutos). Ahora se llega al menú principal, que está compuesto de una lista de media docena de órdenes, y el usuario debe elegir la que desea simplemente pulsando la letra inicial de la orden correspondiente. Por ejemplo, para registrarse como nuevo usuario habría de pulsar la N; para terminar su llamada tendría que digitar la G de *goodbye*. Con la mayoría de las opciones aparecerán nuevos menús, y el usuario podrá ir seleccionando la opción que le interese hasta llegar a lo que desea. El BBS es como un árbol: se empieza por el tronco (el menú principal) y se van escogiendo distintas "ramas" (bifurcaciones) con cada submenú. Prestel, al igual que la mayoría de las otras bases de datos, se vale de la misma idea.

La sección *New user* es para quienes desean inscribirse en el BBS. Al interesado se le solicita su nombre y dirección, y éste puede elegir su propia contraseña para identificarse en futuras llamadas. La orden *Information* (información) le proporciona una detallada relación técnica acerca del sistema. La sección *Utilities* (servicios) le informa de cuánto ha durado su llamada y también le da a conocer detalles relativos a todos los otros BBS.

La sección *Bulletins* (comunicaciones) contiene avisos al público que los usuarios han colocado en el sistema para que los lean otras personas. La sección *Messaging* (mensajes), por su parte, es para que el afiliado al servicio lea mensajes de índole privada que otras personas pueden haber dejado para él. Del mismo modo, puede enviarles mensajes privados a otros usuarios, o a un grupo de usuarios que posean algo en común (por ejemplo, que todos sean propietarios de ordenadores BBC Modelo B). Quizá estas características sean el aspecto más atractivo de los servicios y probablemente a ellas se debe su creciente popularidad.

Un servicio ligeramente diferente, denominado Rewtel, ofrece una base de datos especializada en componentes electrónicos e información relativa a los mismos. También acepta desde el teclado pedidos directos de compra de artículos en existencia, pero sólo a los suscriptores. A diferencia del BBS, el usuario da entrada a "palabras-clave" para especificar su área de interés. Por ejemplo, si deseara utilizar el servicio de compra, debería digitar HELP REWSHOP. También posee una configuración del BBS: dando entrada a la palabra CHALK puede luego dejar un mensaje. Las personas no suscritas al servicio pueden emplearlo: se les permite permanecer ocho minutos en el sistema. Distel y Maptel son servicios similares de base de datos destinados ante todo al pedido de equipos electrónicos e informáticos. La ventaja de estos sistemas comerciales reside en que funcionan las 24 horas del día y no se encuentran ocupados con tanta frecuencia como ocurre con los BBS.

Mientras que los grandes ordenadores de unidad principal han tenido acceso al teléfono desde hace algún tiempo, sólo recientemente, con el descenso de los precios de los modems y la creciente sofisticación de los microordenadores, este tipo de comunicación se ha puesto al alcance de los usuarios de ordenadores personales. En los próximos años, es probable que el modem se convierta en un accesorio tan corriente para el ordenador personal como la impresora o la grabadora de cassette.

Tablón de anuncios computerizado
Una de las aplicaciones más gratificantes del modem consiste en permitir el acceso a un tablón de anuncios electrónico. Éstos son versiones informatizadas de los tradicionales tablones de anuncios existentes en clubs y asociaciones diversas. Los mensajes se pueden "colgar" para que los lea cualquier persona o bien un usuario determinado que posea las contraseñas correctas. Se anuncian reuniones y se ponen en venta equipos de informática de segunda mano. Incluso se pueden cargar juegos u otros listados de programas en un disco o cassette

Editores de pantalla

La mayoría de los microordenadores permite editar programas en la pantalla, ahorrando tiempo y esfuerzo

Todo el mundo comete errores al utilizar el teclado de un ordenador, y por esa sencilla razón es necesario disponer de medios de edición. Son muchas las situaciones en que sería deseable modificar los datos visualizados, no sólo con la finalidad de corregir un error de digitación o rectificar una sentencia errónea, sino de actualizar información.

Muchos programas aplicativos incluyen alguna forma de "editor" especializado, como la posibilidad de eliminar oraciones, trasladar párrafos a otros lugares y reemplazar un nombre o una frase por otros siempre que aparezcan.

No obstante, casi todos los ordenadores personales poseen algún tipo de editor (incorporado en su sistema operativo en ROM), preparado para editar listados de programas. Los programas son sumamente proclives a los errores. En todo programa se producen errores de sintaxis con gran regularidad y, casi con toda certeza, en su funcionamiento habrá defectos que será necesario eliminar, por no hablar de las mejoras que fuera preciso realizar con posterioridad. Las configuraciones que ofrezca el editor de un ordenador pueden suponer una enorme diferencia en cuanto al tiempo de desarrollo de un programa largo. Debemos recalcar, no obstante, que un buen programador invierte una considerable cantidad de tiempo en verificar el funcionamiento de su programa sobre el papel antes de proceder a digitar el listado en un ordenador. Digitar la primera solución que a uno se le pase por la cabeza es un pésimo procedimiento de programación, pues luego se ha de destinar el 90 % del tiempo de desarrollo del programa a eliminar errores.

Existen dos clases de editores: los "editores de pantalla" y los "editores de línea". Los primeros son considerablemente más flexibles y fáciles de utilizar, pero los segundos están mucho más difundidos entre los ordenadores personales. Los editores de línea provienen de la época en que todos los cálculos se efectuaban en un ordenador remoto a través de teletipos o terminales. Los teletipos tenían una memoria buffer de sólo una línea de 80 caracteres (80 bytes). El programador podía obtener una lista impresa de todo el programa digitando LIST, pero si, por ejemplo, se necesitaba efectuar una corrección en la línea 120, se debía volver a digitar entera otra vez esa línea. En algunos sistemas, digitar EDIT 120 resultaría en una salida impresa de esa línea determinada, pudiéndose entonces introducir modificaciones o supresiones mediante las teclas de "retroceso" y "borrar" (las inserciones seguían siendo imposibles). Se agregaron otras órdenes como DELETE (una escala especificada de números de línea), pero aún existía la limitación de tener que llamar y modificar una línea entera.

Los editores de muchos ordenadores personales aún funcionan como si sólo tuvieran un buffer de una línea cuando, de hecho, toda la pantalla es un mapa de memoria: cada localización de carácter corresponde a un byte de memoria.

Un editor de pantalla es mucho más eficaz. Con él se pueden desplazar texto o gráficos por la pantalla con gran comodidad. Cuando se pulsa RETURN, el editor lee al intérprete toda la línea sobre la cual se halla el cursor, aquél la ejecuta (si la línea consiste en una orden) o le da entrada en el programa (si comienza con un número de línea). Empleando las cuatro teclas con flechas, el usuario puede desplazar el cursor hasta cualquier punto del programa, tal como éste se visualice, y después insertar, eliminar o volver a escribir caracteres como desee.

Para obtener la velocidad necesaria, el editor de pantalla se ha de escribir en código de lenguaje máquina y puede incorporar algunas configuraciones verdaderamente útiles. Los mejores editores de pantalla le permitirán «deslizar» un listado hacia arriba o hacia abajo, e insertar o eliminar líneas o caracteres individuales. Algunos disponen de órdenes similares a las de los procesadores de texto, para contabilizar y modificar las instancias en que se produce una determinada serie de caracteres.

Con cada nueva generación de ordenadores los editores ganan en sofisticación y sencillez de uso. Con la introducción del "ratón" (véase p. 296) y del software que imita los procesos manuales para cortar y pegar trozos de texto, se está reduciendo gradualmente el tiempo necesario para editar un listado o un documento en su formato definitivo.

Línea por línea

Las configuraciones de edición de un Sinclair Spectrum son considerablemente mejores que las de la mayoría de los tipos de editores de línea, aunque de ningún modo es tan fácil de utilizar como un editor de pantalla. Para cambiar una línea determinada de un programa, se ha de desplazar la posición normal del señalador (>), que aparece entre el número de línea y la línea en sí, hasta la línea correspondiente, utilizando las teclas arriba y abajo del cursor





Ian McKinnell

Sharp MZ-711

Un ordenador personal con la ventaja de que permite acoplar los periféricos opcionales dentro de su carcasa

El Sharp MZ-711 es una máquina interesante con algunas configuraciones bastante insólitas. En cuanto a construcción, responde al elevado estándar del diseño japonés y da buen resultado. El color es bueno (muy uniforme y brillante) y, aparte de un ligero aspecto borroso al utilizar ciertas combinaciones de colores, produce textos muy legibles.

No obstante, algunos aspectos de su diseño son extraordinarios. El teclado, por ejemplo, podría ser muy satisfactorio si no fuera porque su modalidad es exactamente la contraria de la que uno pudiera esperar: la normal es la de mayúsculas, mientras que las minúsculas se obtienen pulsando las teclas de cambio (SHIFT). Aunque esta disposición, sin duda alguna, es muy cómoda para la programación en BASIC, dificulta notablemente cualquier otra operación Afortunadamente, existe un sencillo procedimiento para hacer reversible la modalidad del teclado de este ordenador: al pulsar las teclas CTRL y E las mayúsculas se transforman en minúsculas, mientras que al digitar CTRL y F se vuelve a la modalidad anterior.

La unidad para cassette del Sharp MZ-711 está concebida como un accesorio opcional que se introduce en la carcasa principal; pero la mayoría de los usuarios la necesitarán. Este sistema es lento pero fiable y todas las cintas se cargan a la primera vez. No obstante, el sistema operativo de la cassette posee varios puntos débiles: sólo existe disponibilidad para los nombres de archivo más elementales y no hay control sobre el motor de la cassette (que hubiera sido muy fácil de incluir). Esto limita la utilidad de la unidad virtualmente a un almacenamiento de un archivo por cinta. Las limitaciones de este sistema son sorprendentes, ateniéndonos a la experiencia de Sharp en otros campos, como el de los equipos de alta fidelidad, en que son habituales sus lotes de cintas controladas por solenoide.

La impresora-plotter incorporada (se trata, nuevamente, de un accesorio opcional que encaja en la carcasa) posee un mecanismo de cabeza de impresión y unidades de lápiz muy frágiles y susceptibles de estropearse. Cuando la máquina no se va a emplear durante un período prolongado, conviene retirar estas últimas y guardarlas. Lamentablemente, son unidades muy pequeñas que se pueden perder con facilidad al quitarlas de la impresora. También tienen cierta tendencia a secarse con rapidez; en



Chris Stevens

El teclado
El teclado del Sharp MZ-711 le confiere a la máquina un aire profesional. Las teclas están correctamente espaciadas, son de tamaño natural, de recorrido total y aptas para el tratamiento de textos. Cinco teclas de función programable, que responden a las teclas de cambio (SHIFT), ofrecen diez funciones y simplifican las operaciones. Los símbolos para gráficos están dispuestos por secuencia lógica, a diferencia de las otras teclas, que responden al formato QWERTY. Ello significa que puede resultar difícil descubrir cuál es la combinación de teclas que producirá el símbolo requerido

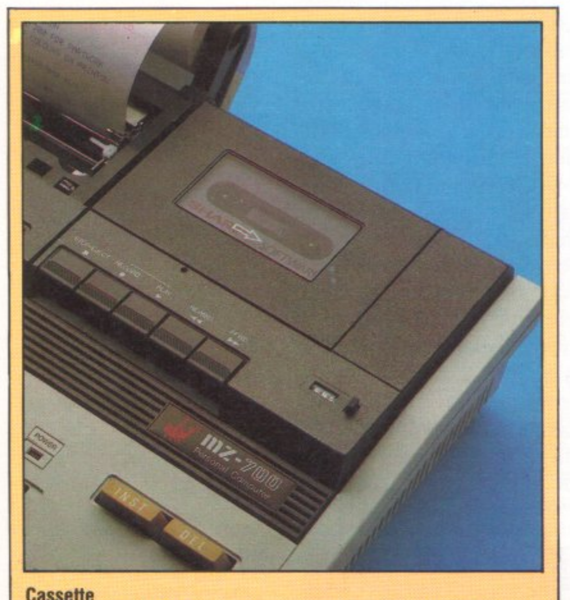
cuanto a la calidad de la impresión, a veces no es del todo satisfactoria.

El intérprete de BASIC se carga de la cinta en vez de estar incorporado en la ROM. Gracias a ello la máquina puede utilizar lenguajes alternativos.

La documentación técnica es de alto nivel. Incluye mapas de memoria y diagramas de circuitos, un listado de lenguaje ensamblador del software incorporado del sistema, así como otros detalles técnicos. No obstante, la documentación carece de diagramas de tiempos, aunque así y todo el manual es mucho mejor que la mayoría de los que acompañan a otros ordenadores.

Los fabricantes del Sharp MZ-711 tienen una inexplicable tendencia a restarles importancia a algunas de sus configuraciones menos usuales. Un ejemplo de esto se produce al ejecutar el programa de demostración. Mientras que el manual da a entender que la máquina puede producir los ocho colores usuales, el programa parece capaz de crear una gama de al menos ocho veces ese número.

Sin embargo, si se inspecciona el programa, se podrá descubrir que en la localización 40960 se ha colocado (POKE) una pequeña rutina en código de lenguaje máquina a la cual se llama en determinados puntos del programa para producir una gama de colores que va de tonos pastel claros a tonalidades oscuras. Resulta difícil determinar la verdadera escala de colores, porque la documentación relativa a las posibilidades de color es más bien modesta.



Cassette
La cassette del MZ-711 es un accesorio opcional, si bien se puede incorporar a la carcasa principal. No obstante, el sistema operativo no posee control sobre el motor de la cassette, lo cual limita sus aplicaciones



Impresora-plotter
Al igual que la cassette, la impresora-plotter es un accesorio opcional. Utiliza cuatro lápices esferográficos en miniatura para producir texto y gráficos en un papel de 11,5 cm. Tres modalidades de funcionamiento permiten tres dimensiones de texto y, en consecuencia, 26, 40 u 80 columnas en el papel

Conector para impresora

Interruptor para la impresora
Selecciona una impresora interna o externa

Conector de potencia

Conector para el altavoz

ROM
El sistema de ROM contiene sólo un minúsculo programa denominado monitor, que puede manipular posiciones de memoria o cargar el intérprete de BASIC desde la cinta

RAM
Ocho chips contienen los 64 K de RAM estándar

Chip de reloj-sincronización

Adaptador de interface para periférico
Este PIA (Peripheral Interface Adaptor) controla la interface del teclado y de la cassette

Conector para teclado

Mando de borrado

Control de volumen

CPU

Bus de ampliación Z80
Para ampliación de memoria y periféricos

Conexión para impresora externa
Se trata de una interface en paralelo, pero, como no responde al estándar Centronics, sólo funciona con impresoras Sharp

Conectores para cintas externas

Conectores para palanca de mando

Conector de cinta interna

Conector para video
Proporciona las señales para un monitor en color y, mediante el modulador de RF incorporado en la carcasa, para un televisor

Interruptor de cassette
Si su cassette se negara a cargar los programas, este interruptor alteraría la fase de la señal, con lo cual se debería eliminar el problema

Controlador CRT
Este chip diseñado a medida controla todos los aspectos relativos al video

Generador de caracteres
Esta ROM almacena los patrones de bits para los caracteres y símbolos de gráficos

RAM de pantalla

Chris Stevens

SHARP MZ-711

DIMENSIONES
440 × 305 × 86 mm

CPU
Z80A

VELOCIDAD DEL RELOJ
4 MHz

MEMORIA
ROM, 12 K; RAM, 64 K

VISUALIZACION EN VIDEO
24 líneas de 40 caracteres, o gráficos pixel de baja resolución (50 × 80), 8 colores nominales y un número no especificado de tonalidades, con fijación independiente de fondo y primer plano; 127 caracteres predefinidos y 127 caracteres definibles por el usuario

INTERFACES
RS232 en serie, cassette, bus de sistema, impresora en paralelo

LENGUAJE SUMINISTRADO
BASIC en cinta de cassette

OTROS LENGUAJES DISPONIBLES
PASCAL, ASSEMBLER, FORTRAN, FORTH, DAFMOD, TOOLKIT

VIENE CON
Manuales de instalación y de BASIC, cable para TV, cable para cassette, cassette de BASIC y cassette de demostración

TECLADO
Teclas de recorrido total y cinco teclas de función programable con valores con o sin cambio (SHIFT)

DOCUMENTACION
Es muy amplia y cubre importantes detalles técnicos. Un tanto pesada, constituye, no obstante, una buena introducción a la máquina

Sonidos sencillos

La generación de sonido del Spectrum de Sinclair

El Spectrum se está convirtiendo rápidamente en el ordenador personal más popular. Incorpora excelentes medios para gráficos en color y útiles opciones de memoria por un precio reducido. Lamentablemente, se han sacrificado algunas configuraciones con el fin de no encarecer la máquina. La mayoría de las quejas se refieren al teclado y a la utilización de un BASIC no estandarizado, pero tal vez su mayor debilidad radique en su capacidad para sonido. El Spectrum proporciona los elementos esenciales para la generación de sonido; no obstante, produce música mediante un único oscilador de tipo de "pulso". Se puede controlar la duración de una nota y su altura, pero es imposible alterar el tono de una nota ni modificar su envoltura (véase p. 276). La salida es estándar, y se efectúa a través de un altavoz piezoeléctrico interno muy pequeño, que resta musicalidad al sonido. No obstante, Sinclair ha proporcionado salidas de señal alternativas en las conexiones para cassette *mic* o *ear*, aptas para amplificación externa mediante un sistema de alta fidelidad. La capacidad de sonido del Spectrum ofrece la ventaja de la simplicidad de las órdenes de BASIC BEEP y PAUSE, que están íntimamente relacionadas y que le permiten al usuario comprender con mayor facilidad los principios del sonido generado por ordenador. Además, la máquina posee una escala de extensión muy notable: 10 octavas.

Antes de que se pueda producir algún sonido con el Spectrum, primero es necesario conectarlo a un equipo de alta fidelidad o bien hacer audible el *beeper* interno dando entrada a la siguiente orden directa antes de proceder a ejecutar (RUN) un programa de sonido:

```
POKE 23609,100
```

Por otra parte, es preciso advertir que esta orden también aumenta el volumen de la realimentación "click" que se produce en el momento en que se pulsa una tecla en el teclado.

Control de sonido

Para instruir al ordenador para que dé salida a una nota determinada, se utiliza la orden BEEP del siguiente modo:

```
BEEP d,n
```

donde d representa la duración de una nota y n, la altura de ella. El valor de duración se puede establecer entre 0,00125 y 10 segundos; la altura se expresa como el número de semitonos desde *do mayor* (valor 0) de la escala entre −60 y 69. Por ejemplo, la sentencia siguiente toca la nota *la* en 440 Hz, que está a nueve semitonos de *do mayor*, durante medio segundo:

```
BEEP 5,9
```

Para tocar toda una serie de notas separadas por un espacio medido con exactitud, podemos utilizar la orden PAUSE:

```
PAUSE ms
```

Imágenes elementales

Las capacidades para gráficos del Commodore Vic-20

El Commodore Vic-20, al igual que los otros ordenadores personales de la firma Commodore —el 64 y el PET—, está muy bien construido, pero sus fabricantes no se han prodigado en el juego de instrucciones en BASIC de la máquina. En efecto, el usuario del Vic-20 no dispone de órdenes especiales para gráficos, y para tal fin deberá tener un gran conocimiento acerca del funcionamiento interno del ordenador, o bien tendrá que adquirir alguno de los accesorios diseñados para hacer que la programación de gráficos resulte más sencilla.

El Vic-20 dispone de dieciséis colores y cada cuadrado de caracteres puede contener cuatro colores. La visualización en pantalla consta de 23 filas y 22 columnas de celdas de caracteres de ocho pixels por ocho, pero los caracteres también se pueden visualizar en un formato rectangular de 16 por 8.

Capacidad en baja resolución

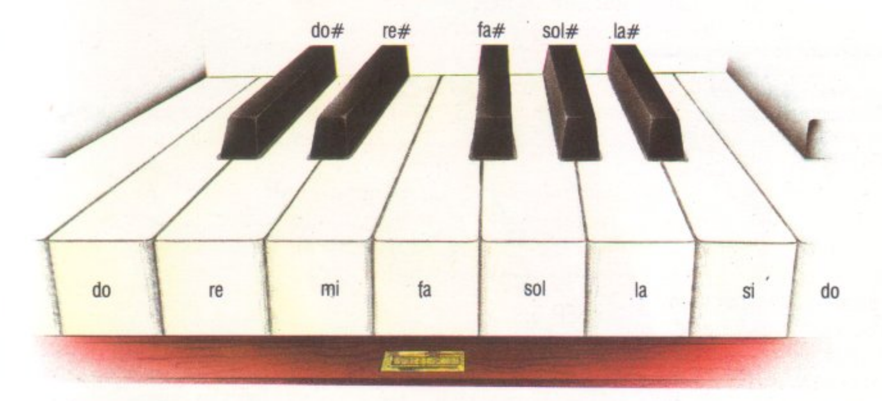
Dispone de caracteres alfabéticos en mayúsculas y en minúsculas, así como de más de 60 caracteres para gráficos PET especiales. Muchas de las teclas del Vic-20 llevan marcados dos pequeños cuadrados en su parte anterior, cada uno de los cuales visualiza un patrón determinado. Además de los cuadrados de medio carácter y de un cuarto de carácter, hay símbolos para jugar a los naipes, diseños de tableros de damas, círculos y numerosos símbolos que se pueden combinar en la pantalla para trazar gráficos, componer tablas, producir



donde ms representa el tiempo en unidades de 0,001 segundo (milisegundos). Para tocar una octava en una escala de *do mayor (do, re, mi, fa, sol, la, si, do)* desde el *do* normal durando cada nota medio segundo y dejando, además, una pausa (PAUSE) de un cuarto de segundo entre cada nota y la sucesiva, cabe utilizar este formato:

```
10 FOR I = 1 TO 8
20 READ N
30 BEEP .5,N
40 PAUSE 250
50 NEXT I
60 DATA 0,2,4,5
70 DATA 7,9,11,12
```

Este programa es una buena ilustración del formato de una escala. Una octava se extiende desde la nota base (en este caso *do mayor*) hasta la siguiente nota con el mismo nombre (el siguiente *do* citado) y abarca 12 semitonos. Se denomina octava porque se compone de ocho notas en una escala mayor.



do mayor

Escalas y pianos

Utilizando la orden INKEY$ se puede emplear el teclado del Spectrum como aproximación al teclado de un piano:

```
 10 REM**************
 20 REM *OCTAVA PIANO*
 30 REM**************
 40 IF INKEY$ = "Q" THEN BEEP 1,0
 50 IF INKEY$ = "2" THEN BEEP 1,1
 60 IF INKEY$ = "W" THEN BEEP 1,2
 70 IF INKEY$ = "3" THEN BEEP 1,3
 80 IF INKEY$ = "E" THEN BEEP 1,4
 90 IF INKEY$ = "R" THEN BEEP 1,5
100 IF INKEY$ = "5" THEN BEEP 1,6
110 IF INKEY$ = "T" THEN BEEP 1,7
120 IF INKEY$ = "6" THEN BEEP 1,8
130 IF INKEY$ = "Y" THEN BEEP 1,9
140 IF INKEY$ = "7" THEN BEEP 1,10
150 IF INKEY$ = "U" THEN BEEP 1,11
160 IF INKEY$ = "I" THEN BEEP 1,12
170 GOTO 40
```

En un programa como éste se pueden introducir muchas mejoras con el fin de crear un "instrumento" de teclado más eficaz. El Spectrum, si bien sus configuraciones sirven sólo para generar sonidos más bien sencillos, puede constituir, sin embargo, un recurso auxiliar muy útil para enseñar música. Por otra parte, es posible conseguir un sonido de mayor sofisticación si el usuario adquiere hardware adicional para generación de sonido que sea compatible con la máquina.

grandes rótulos y crear otros efectos. Los caracteres se pueden visualizar invertidos (negro sobre blanco, en lugar de blanco sobre negro), aumentando las posibilidades. Con paciencia e imaginación se producen visualizaciones de gran calidad.

Los caracteres se pueden hacer aparecer en la pantalla mediante la utilización de la sentencia PRINT o bien a través de (POKE) los códigos requeridos en la pantalla del Vic-20 y las memorias de color. La versión Commodore de la sentencia PRINT es muy eficaz, ya que permite que el usuario defina individualmente el color de cada carácter. Desde dentro de la sentencia PRINT también se puede controlar el movimiento del cursor para producir con facilidad visualizaciones móviles. Colocar (POKE) los caracteres en la pantalla no es tan rápido como imprimirlos (PRINT), pero este procedimiento puede ser muy útil en ciertas circunstancias.

Capacidad en alta resolución

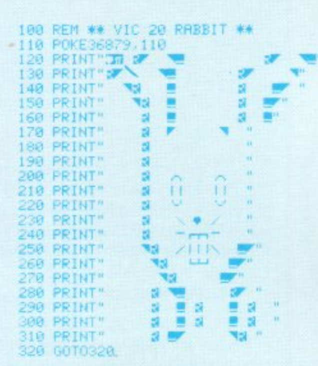
En un Commodore Vic-20 sin ampliar se pueden obtener gráficos de alta resolución, pero la memoria disponible sólo es suficiente para utilizar la mitad de la pantalla. El proceso por el cual se puede obtener una alta resolución se denomina "mapa de bits", técnica que le permite al programador controlar cada pixel individual dentro de una superficie determinada de la pantalla. Cada celda de caracteres de la cuadrícula de 23 filas por 22 columnas se compone de 64 pixels dispuestos en ocho filas de ocho. Existe una relación matemática entre las coordenadas dadas de un punto de pixel (x,y) y el bit correspondiente en la matriz del carácter. Ésta se puede emplear, junto con una combinación de órdenes POKE, para producir visualizaciones compuestas por pixels individuales.

Cartucho de superampliación

Producir gráficos de alta resolución mediante mapas de bits puede ser un proceso largo y difícil. Una solución alternativa es comprar el cartucho Super Expander de Commodore, que le propociona al usuario órdenes en BASIC para alta resolución, música y color. Las órdenes para alta resolución incluyen GRAPHIC, para establecer la modalidad de visualización, POINT, para trazar un punto de pixel en la pantalla, y PAINT.

La utilización del cartucho entraña dos inconvenientes principales: no existe orden UNPOINT para borrar el punto de pixel, y no se puede pintar (PAINT) a ambos lados de una línea en diagonal sin alterar la línea en sí misma.

Los gráficos de baja resolución del Vic-20 están bien diseñados y son muy flexibles, pero los gráficos de alta resolución resultan difíciles de utilizar sin adquirir un cartucho enchufable.



Un conejo programado
Este listado de programa para el Vic-20 demuestra la versatilidad del juego especial de caracteres de Commodore. La figura del conejo está realizada enteramente a partir de este juego de caracteres predefinidos. Resulta más sencillo ver exactamente cómo está construida haciendo mover el cursor por la visualización en pantalla.

Los brazos-robot

Estos precisos brazos mecánicos mediante una interface se pueden conectar a cualquier ordenador personal con una puerta en paralelo

¿Ha deseado alguna vez que de algún modo el ordenador que utiliza pudiera llevar a cabo una tarea tan sencilla como preparar una taza de té? No existe problema alguno, con una interface apropiada, en programar a un ordenador para que encienda y apague una tetera. Pero cuando llega el momento de manipular objetos físicamente, como inclinar la tetera para verter agua caliente dentro de una taza, entonces se necesita un brazo mecánico. Recientemente este tipo de dispositivos (denominados *brazos-robot*) ha salido al mercado y está a disposición de los usuarios de ordenadores personales. Son pequeñas versiones de los brazos industriales que emplean empresas como Seat, Renault y otras para realizar los trabajos de soldadura y pintura en sus cadenas de montaje de coches. El "Armdroid" de Colne Robotics, que probablemente fue el primer brazo-robot apto para utilizar con un ordenador personal, salió al mercado en 1981. Aunque el brazo no es movible (a menos que se lo monte sobre un robot móvil), permite manipular objetos con un notable grado de precisión.

Los principales componentes del brazo-robot, aparte de las secciones metálicas propiamente dichas, son los motores paso a paso, que facilitan el movimiento de las secciones según cálculos exactos. Hay seis motores: uno para hacer rotar el brazo por la "muñeca", otro para controlar la articulación del "hombro", un tercero para la articulación del "codo", y tres para dirigir el movimiento de la "mano". Todos estos motores se pueden controlar fácilmente mediante un ordenador.

Todo lo que se requiere para conectar el brazo a un ordenador es una puerta en paralelo de ocho bits. Un bit determina si la información pasa hacia o desde el robot. Tres bits de dirección se utilizan para seleccionar el motor deseado, y los otros cuatro controlan la dirección y la velocidad del movimiento. También se envían señales de reloj para sincronizar los movimientos del brazo robot con las instrucciones del ordenador. Para acelerar el proceso y permitir que el brazo realice maniobras complejas, hay unos cerrojos electrónicos incorporados en el sistema de circuitos que permiten que una

Bobinas
Las bobinas de cuerda están dispuestas de tal modo que si se modifica el ángulo del hombro, el ángulo del codo cambiará automáticamente, para mantener el "antebrazo" en el mismo ángulo respecto a la horizontal

Codo
Posee una libertad de movimiento de 270°

Brazo superior

Antebrazo

Compensador de tensión
Esta polea asegura que los tres "dedos" ejerzan la misma presión sobre el objeto que asen, aun cuando éste sea de forma irregular

Muelle de tensión
Todos los movimientos se transfieren desde los motores al brazo mediante muelles, y para asegurar la precisión éstos se han de mantener tensados

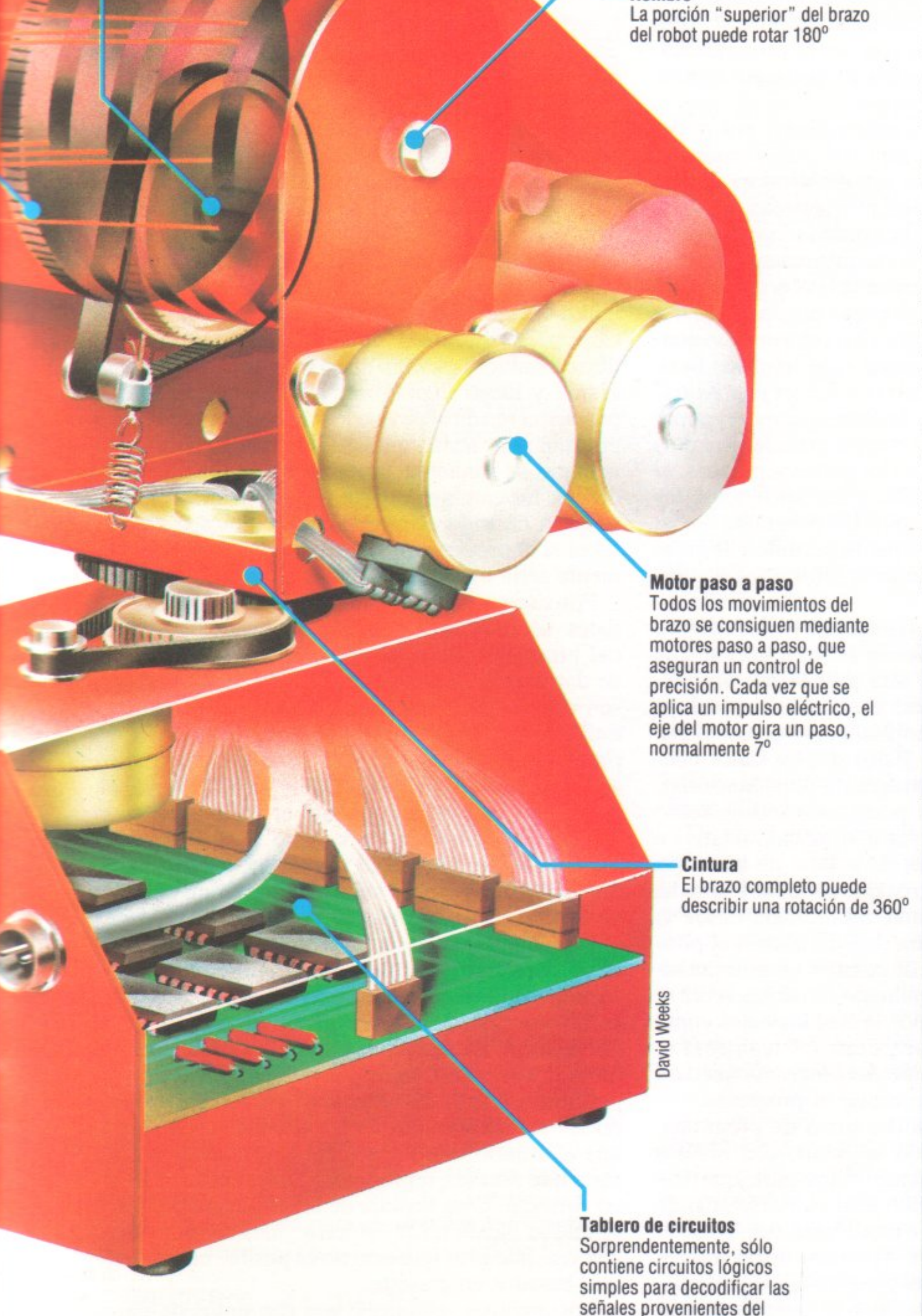
Mano
Los tres "dedos" de la mano-pinza tienen articulaciones de bisagra unidas por muelles y rellenas de goma para ayudar a asir los objetos cuando no existen sensores

Muñeca
La muñeca tiene una inclinación máxima de 180° y también puede describir una rotación completa de 360°

Interface
El brazo se puede conectar en cualquier puerta en paralelo de ocho bits. Tres bits se utilizan para indicar qué motor se está direccionando, uno especifica si los datos se están recibiendo o si se están enviando, y cuatro son para la información

combinación de motores opere simultáneamente, "reteniendo" la información de un motor mientras se está instruyendo a los otros.

Para hacer que el brazo se coloque en posición y agarre un objeto, primero es necesario dividir el movimiento global en una serie de pasos simples. Cada motor habrá de ser instruido por separado para un movimiento preciso que, conjuntamente con los demás, compondrá el movimiento total del brazo-robot. Esta información se almacena luego en la memoria del ordenador y se puede hacer que el brazo repita la operación tantas veces como sea necesario.



Si el brazo estuviera manipulando objetos delicados (las pruebas se suelen realizar con un huevo), se ha de hacer que el ordenador controle la presión de la pinza. Si ésta es escasa, el huevo caerá; si es excesiva, se romperá la cáscara. Para transmitir la información desde el brazo al ordenador se utilizan diversos procedimientos, pero el más común se basa en microinterruptores sencillos. Éstos se pueden acoplar para que establezcan los límites del desplazamiento del brazo (la mayoría de los brazos de bajo costo no incluyen sensores), o se pueden incorporar en la pinza para detectar un límite de presión preestablecido.

De los sistemas alternativos de interruptores que se emplean en la mayoría de los brazos más grandes, el principal se basa en la detección de presión. La resistencia eléctrica de ciertos materiales se altera cuando éstos se someten a un cambio de presión, y estas fluctuaciones se pueden medir. Aunque este método es más caro, proporciona resultados de mucha precisión.

Si el programa no admite realimentación de información del brazo al ordenador, se dice que es un "bucle abierto" o determinista. Si existe alguna forma de realimentación que ajuste las acciones ejecutadas bajo el control del programa, entonces el sistema es de "bucle cerrado" o estocástico. Aquí los interruptores o sensores de presión se utilizan para limitar el cierre de la pinza en un punto en el cual el huevo se sostiene con firmeza pero sin que se rompa.

Muchos de los sistemas de robot más sofisticados incluyen múltiples sensores para medir la luz, el calor y otras variables. Estos sensores se pueden emplear para llevar el registro de lo que sucede mientras el brazo lleva a cabo su tarea y para informar si algo no funciona bien: por ejemplo, que un robot soldador está practicándose agujeros a sí mismo.

Ian McKinnell

Manejo del lenguaje
Escribir un programa para controlar un brazo robot es relativamente sencillo. En BASIC, la principal tarea sería la de lograr que el ordenador aceptara las órdenes de control provenientes del teclado y que las pasara al brazo a través de la puerta valiéndose de POKE. De la misma manera, se debería poder leer la entrada desde el brazo mediante la función PEEK por la puerta relacionada. Si el principal requerimiento es el relativo a la velocidad, entonces es esencial la programación en código de lenguaje máquina. El FORTH es un lenguaje que ofrece la facilidad de programación del BASIC y gran parte de la velocidad asociada al código de lenguaje máquina. A las secciones de un programa, como las subrutinas o procedimientos, se les dan nombres que se pueden incorporar al juego de órdenes del lenguaje. Ello consigue que este lenguaje sea sumamente eficaz para aplicaciones especializadas, como lo son los programas de control para brazos-robot.

Ampliación de archivos

Proseguimos con nuestro proyecto de programación dando una mirada a la manipulación de archivos

El programa para la agenda de direcciones que hemos venido desarrollando en los últimos capítulos de nuestro curso de programación BASIC es, en realidad, un tipo de base de datos sencillo y, como tal, implica el concepto de "archivos". Esta palabra se emplea de varias maneras que, aunque ligeramente diferentes, están relacionadas entre sí. Empezaremos por analizarlas con cierta profundidad, de modo que de aquí en adelante podamos utilizar la palabra con mayor precisión.

En la programación de ordenadores, nos podemos hacer la idea de que un "archivo" se asemeja mucho a los archivos de un archivador. Es un conjunto de informaciones relacionadas entre sí que se almacenan juntas. Los ordenadores almacenan los archivos en discos o en cintas magnéticas. A cada "archivo" de información se le otorga un nombre propio para que el ordenador pueda acceder a él cada vez que sea necesario. La información contenida en una cinta de cassette o en un disco flexible puede ser un programa o pueden ser los "datos" que utiliza un programa. Tomando como ejemplo la agenda de direccciones computerizada, la información necesaria consta de dos partes separadas: el programa propiamente dicho y los datos sobre los cuales trabaja el programa. El programa es el conjunto de instrucciones que le permiten al ordenador (y al usuario) manejar la información y trabajar con ella.

Los datos que emplea el programa son el conjunto de registros que contienen la información que usted esperaría hallar en una agenda de direcciones: nombres, direcciones, etc. También incluye cierto tipo de datos que normalmente no están disponibles para el usuario. Éstos son los datos "domésticos" que utiliza el programa para funcionar. Entre este tipo de datos podríamos incluir como ejemplos: las "banderas"; la información relativa a las dimensiones habituales de la base de datos (es decir, el número de registros que contiene); si se ha llevado a cabo o no alguna clasificación desde la última vez que se insertó un registro nuevo; o, posiblemente, una indicación de cuántas veces se ha accedido a un registro determinado o cuántas veces se lo ha impreso. La razón por la cual los datos como éstos (y los datos que componen los registros) se han de tratar separadamente del programa quedará clara apenas intentemos ejecutar el programa.

En capítulos anteriores del curso de programación BASIC hemos utilizado las sentencias READ y DATA como formas de poner datos para que trabajen en un programa. Esto sólo es viable cuando los datos no están sujetos a modificaciones, como el número de días de un mes. Si los datos son susceptibles de modificaciones, el programa puede prepararse para ello en la pantalla y se pueden utilizar INPUT, INKEY$ u otros métodos para traspasarle los datos al programa. Un ejemplo del uso apropiado de esta forma de entrada de datos podría ser un juego de adivinanza de números, en el cual una parte del programa podría asumir la forma:

```
PRINT "ADIVINE EL NUMERO"
INPUT N
IF N<> COMPNUM THEN...
```

Los datos del programa de la agenda de direcciones están sujetos, sin embargo, a considerables modificaciones. En teoría, todos los registros se podrían almacenar en el programa y se los podría leer en matrices apropiadas utilizando sentencias READ y DATA. Pero después, a todos los datos de que constaran los registros se les habría de dar entrada como parte del programa. Cada vez que se efectuaran cambios (agregar o eliminar nombres y direcciones, por ejemplo) sería necesario introducir considerables alteraciones en el propio programa. Como mínimo, ello significaría imprimir el programa, verificarlo para determinar dónde sería necesario introducir cambios, escribir nuevos segmentos del programa y luego digitarlos. No obstante, el mayor problema residiría en que los segmentos nuevos del programa no serían módulos completos de programa que se pudieran probar de manera independiente; los cambios estarían diseminados al azar a lo largo de todo el programa. La única forma de saber si el programa modificado funciona correctamente sería ejecutarlo y ver qué sucede.

Por suerte, nada de esto es necesario porque los datos se pueden almacenar independientemente del programa. Esto se consigue creando archivos de datos en la cassette o en el disco. Estos archivos son conjuntos de registros que se tratan de forma muy similar a los datos de una sentencia DATA. El programa es capaz de "abrir" uno o más de estos archivos, leer los datos de ellos (por lo general, en una matriz) y después "cerrar" el archivo. Cuando se necesita introducir un cambio en los datos, el programa abre el archivo adecuado, lee los datos, los modifica y después vuelve a escribir en el archivo los datos modificados.

Con los sistemas de ordenadores basados en disco, localizar un archivo determinado y leerlo o escribir en él es bastante rápido: la localización del archivo se realiza en una fracción de segundo y las operaciones de lectura o escritura por lo general tardan, a lo sumo, unos pocos segundos. Por otra parte, un sistema de ordenador basado en cassette puede ser mucho más lento; además, es probable que el usuario tenga que rebobinar la cinta y esperar a que ésta sea reproducida hasta hallar el archivo correcto. Otra ventaja de la utilización del disco es que se puede tener "abierto" más de un archivo a la vez, mientras que esto no es posible en los sistemas basados en cassette.

Los archivos, entonces, son conjuntos de datos almacenados en un medio de almacenamiento de gran capacidad y son aptos para que los emplee un programa o varios. A un programa de tratamiento

de textos, por ejemplo, se le podría dar acceso al mismo conjunto de nombres y direcciones para que escriba automáticamente cartas "personalizadas".

Los archivos se manipulan de diferentes maneras a tenor de la versión de BASIC que se utilice. La mejor forma de saber cómo los maneja su ordenador consiste en ver lo que dice el manual acerca de las sentencias OPEN y CLOSE ("abrir" y "cerrar", respectivamente) y probar con algunos ejemplos. La descripción que le estamos ofreciendo aquí es muy general e intenta proporcionarle una idea global acerca de la utilización de los archivos.

Los archivos pueden ser secuenciales o de acceso directo. En un archivo secuencial, la información se almacena con el primer dato en primer lugar, el segundo a continuación, en seguida el tercero, etc. Un archivo de acceso directo, por el contrario, está organizado de modo que el ordenador pueda remitirse directamente a la información requerida, sin tener que comenzar por el principio e ir revisando los datos hasta localizar la información requerida. Un archivo secuencial es como ver una película: uno comienza desde el principio y la va mirando hasta el final. En cambio, si usted mirara la película en su casa con un equipo de video, sería en cierta manera como un archivo de acceso directo: podría hacer avanzar o retroceder la cinta y, de esa manera, contemplar exclusivamente la secuencia que le interesara. Nosotros sólo vamos a tener en cuenta los archivos secuenciales, porque son mucho más adecuados para los sistemas de cassette.

Supongamos que desea llevar un registro de la media de las temperaturas diarias durante la semana. Éstas podrían ser:

LUNES	13.6
MARTES	9.6
MIERCOLES	11.4
JUEVES	10.6
VIERNES	11.5
SABADO	11.1
DOMINGO	10.9

Para mantenernos en un nivel simple, todos estos datos se tratarán como datos numéricos, con el lunes como día 1 y el sábado como día 7. Entonces la información se puede representar así:

1,13.6,2,9.6,3,11.4,4,10.6,5,11.5,6,11.1,7,10.9

Para almacenar todos estos datos en un archivo secuencial, en el programa serán necesarios los siguientes pasos:

OPEN el archivo
Escribir los datos en el archivo
CLOSE el archivo

Siempre que se utilice la sentencia OPEN será necesario aclarar si vamos a escribir en el archivo datos del ordenador (una salida) o si vamos a leerle al ordenador datos del archivo (una entrada). En el BBC Micro, esto se realiza utilizando las sentencias OPENOUT y OPENIN. Sus equivalentes en BASIC Microsoft son OPEN "0" y OPEN "I". Un breve fragmento de programa para escribir los datos anteriores en un archivo (en BASIC Microsoft) podría ser:

```
100 OPEN "0", # 1, "DAT.TEMP"
110 PRINT # 1,1,13.6,2,9.6,3,11.4,4,10.6,5,
    11.5,6,11.1,7,10.9
120 CLOSE # 1
```

La palabra OPEN de la línea 100 hace que el archivo quede disponible para el programa. OPEN va seguida de "0" para indicar que los datos saldrán del programa para ser almacenados en el archivo. Luego sigue # 1, que le dice al ordenador que en nuestro programa nos referiremos a éste como el archivo número 1. A cada archivo se le da un número convencional que desde ese momento se utilizará con las sentencias INPUT # o PRINT # cuando deseemos leer o escribir datos en él. Por último, tenemos el nombre del archivo entre comillas dobles. Hemos denominado DAT.TEMP a nuestro archivo para indicar que contiene temperaturas y no es un programa sino un archivo de datos.

A continuación escribimos un programa completo en BASIC Microsoft para dar entrada a los datos en un archivo y posteriormente leerlos e imprimirlos:

```
100 OPEN "0", # 1, "DAT.TEMP"
110 PRINT # 1,1,13.6,2,9.6,3,11.4,4,10.6,5,
    11.5,6,11.1,7,10.9
120 CLOSE # 1
130 REM LAS LINEAS 130 y 140 SON LINEAS
    "FICTICIAS" PARA
140 REM REPRESENTAR EL PROGRAMA
    INTERPUESTO
150 OPEN "I", # 1, "DAT.TEMP"
160 FOR X = 1 TO 7
170 INPUT # 1, DIA, TEMP
180 PRINT "DIA";DIA, TEMP
190 NEXT X
200 CLOSE # 1
210 END
```

Éste abre un archivo, numerado # 1 y denominado DAT.TEMP, escribe datos en él utilizando la sentencia PRINT # y luego cierra (CLOSE) el archivo. Más adelante en el programa se abre el mismo archivo utilizando tanto el número del archivo como su nombre (el número no ha de ser necesariamente el mismo que se empleó al crear el archivo, pero el número usado en las sentencias PRINT # o INPUT # debe ser el mismo que se le asignó al nombre del archivo cuando se abrió éste). En la línea 170, INPUT # 1 indica que la entrada provendrá de un archivo numerado # 1 (el archivo DAT.TEMP) y no desde el teclado.

De momento vamos a dejar el tema de la manipulación de archivos y volveremos al programa de la agenda de direcciones y a algunos de los componentes incluidos en la subsección INICIALIZACION del programa. Primero, observemos la cantidad de espacio de memoria que se necesita para un único registro en el archivo de la agenda de direcciones (aquí la palabra "archivo" se está empleando en el sentido de base de datos, o sea, conjunto de todos los registros relacionados, y no en el sentido de sistema operativo).

La utilización de campos de longitud fija hace que en cierto modo se desaproveche espacio de memoria, pero simplifica de manera considerable la programación. Si dejamos una línea entera para cada campo, con 40 caracteres por línea, todos ellos se guardarán en una matriz aun cuando la línea estuviera compuesta básicamente de espacios en blanco. No obstante, en algunas versiones de BASIC, cuando las matrices se DIMensionan, cada elemento puede tener hasta 256 caracteres de longitud. El dimensionamiento establece tan sólo el número de

elementos de la matriz, no las dimensiones de cada uno de los elementos.

Si su ordenador posee un BASIC capaz de manipular matrices multidimensionales, se podría utilizar una dimensión separada para cada uno de los campos; pero muchas versiones de BASIC no pueden hacerlo, de modo que investigaremos aproximaciones alternativas. El método más sencillo consiste en emplear una matriz separada para cada uno de los campos. Pero aquí hay una manera de "hacer trampa" si desea utilizar matrices multidimensionales y su BASIC no pudiera manipularlas.

El truco consiste en tratar a todos los elementos que estarían en la matriz multidimensional como si fueran los elementos de una matriz unidimensional. Por ejemplo, una matriz bidimensional con tres filas y cinco columnas se dimensionaría así: DIM A(3,5) y contendría un total de 15 elementos: de A(1,1) a A(3,5). La misma información se podría retener en una matriz subíndice corriente como ésta: DIM A(15). Siempre que una matriz bidimensional fuera referida como A(F,C) (fila,columna), nosotros la sustituiríamos por A((F – 1)*5 + C).

En caso de emplear una matriz separada para cada campo, hemos de decidir cómo DIMensionarlas. La forma más sencilla es utilizar un tamaño de matriz fijo, pero ello limita el número total de registros que podemos almacenar en la base de datos. Una medida más atinada sería establecer las dimensiones de la matriz de acuerdo a cuantos registros haya en uso. No obstante, no todas las versiones de BASIC admiten que las matrices sean todo lo grandes que el usuario desearía. Aun cuando lo hicieran, un elevado número de registros en la base de datos ocuparía muy pronto toda la memoria disponible en el ordenador. He aquí un programa que le permitirá averiguar el número máximo de elementos que admite el ordenador que emplea usted. Sin embargo, muchas versiones de BASIC permiten que en una matriz haya tantos elementos como se desee, exactamente hasta el punto de ocupar toda la memoria disponible. Cada vez que el programa le pregunte "¿QUE TAMAÑO DE MATRIZ?" dé entrada a un valor mayor, hasta que al fin obtenga un mensaje de error. El CLEAR de la línea 100 tiene el efecto de eliminar la matriz al final de cada pasada. Sin esta sentencia, se obtendría un mensaje de error en la línea 30 al volver a dimensionar una matriz.

```
10 READ D$
20 INPUT "¿QUE TAMAÑO DE MATRIZ?";A
30 DIM N$(A)
40 FOR L = 1 TO A
50 LET N$(L) = D$
60 NEXT L
70 FOR L = 1 TO A
80 PRINT L,N$(L)
90 NEXT L
100 CLEAR
110 GOTO 10
120 DATA "MI COMPUTER"
130 END
```

Aun cuando en cada elemento sólo se admitieran 40 caracteres, con cinco campos por registro, y si hubiera 256 elementos apartados para cada matriz, la cantidad de memoria que se necesitaría para retener todos los datos en la memoria principal sería enorme. Así, si se requiriera un byte por cada carácter a almacenar, necesitaríamos 51 200 bytes (5 × 40 × 256 bytes) sólo para los datos. Obviamente, no es práctico ocupar tanta memoria principal con la información, y ése es el motivo por el cual lo más indicado es emplear archivos de datos separados.

Lamentablemente, como ya hemos sugerido, las rutinas para manipulación de archivos pueden resultar algo difíciles de utilizar. Si no deseamos emplear archivos externos, la única alternativa es colocar los datos en una sentencia DATA de modo que siempre esté presente en el programa. No obstante, una vez haya probado con algunos programas cortos tanto para escribir datos en archivos externos como para tomarlos, todo el proceso le resultará mucho más claro y fácil de entender. A modo de ilustración, hemos elegido dos máquinas y dos versiones de BASIC muy diferentes para complementar el corto programa para la temperatura diaria que dimos en BASIC Microsoft.

Estas versiones son para el Sinclair Spectrum y el BBC Micro; ambas difieren considerablemente de nuestro habitual BASIC Microsoft y para obtener detalles relativos a alguna de estas diferencias los lectores han de remitirse a los recuadros "Complementos al BASIC" de anteriores capítulos del curso de programación BASIC.

En la versión del Spectrum, OPEN # y CLOSE # están reservadas para emplear con microdisco. Cuando se utiliza almacenamiento en cassette, se necesitan versiones especiales de las órdenes SAVE y LOAD. La orden SAVE corriente se emplea para almacenar programas y variables de programas en cinta (y datos normales en sentencias DATA). Las matrices se pueden guardar en cinta usando la sentencia SAVE-DATA. Esta asume la forma siguiente:

SAVE nombre del archivo DATA nombre de la matriz ( )

El nombre del archivo es el nombre dado al archivo (DAT.TEMP, en el programa en Microsoft). El nombre de la matriz corresponde, simplemente, a éste seguido de un par de paréntesis que encierran un espacio en blanco. Para guardar (SAVE) los resultados de la temperatura diaria, primero deberíamos crear una matriz DIMensionada y escribir los datos en ella, empleando sentencias READ-DATA. Para hacer más evidente la diferencia entre el nombre del archivo y el nombre de la matriz, a la matriz la denominaremos c$ y el nombre del archivo será "DATTEMP".

```
10 DIM c$(14)
20 FOR x = 1 TO 14
30 READ c$(x)
40 NEXT x
50 DATA 1,13.6,2,9.6,3,11.4,4,10.6,5,11.5,
   6,11.1,7,10.9
60 SAVE "DATTEMP" DATA c$( )
70 STOP
80 LOAD "DATTEMP" DATA c$( )
90 FOR L = 1 to 14 STEP 2
100 PRINT "DIA";c$(L,c$(L + 1))
110 NEXT L
120 STOP
```

La línea 60 guarda todos los datos de la matriz c$ en un archivo de datos denominado DATTEMP. El programa se detendrá en la línea 70 y deberá rebobinar la cinta. La tecla CONT reanudará la ejecución del programa. La línea 80 invierte el proceso y almacena los datos de DATTEMP en la matriz c$.

El BBC Micro posee una de las versiones de BASIC más sofisticadas de que disponen los ordenadores personales. Admite la programación estructurada con configuraciones tan avanzadas como una construcción REPEAT-UNTIL y "procedimientos". Antes de que se pueda utilizar un procedimiento, primero éste se ha de definir; más abajo veremos cómo se realiza esto en la versión del programa para el BBC. El BASIC de este ordenador define la dirección del flujo de datos en la sentencia de "apertura", usando sea OPENOUT u OPENIN:

```
10 DIM c$(2,7)
20 FOR F = 1 TO 7
30    FOR C = 1 TO 2
40       READ C$(C,F)
50    NEXT C
60 NEXT F
70 DATA 1,13.6,2,9.6,3,11.4,4,10.6,5,11.5,
   6,11.1,7,10.9
80 INPUT "DIGITE S PARA GUARDAR DATOS",K$
90 IF K$<> "S" THEN GOTO 80
100 PROCSAVE: CLEAR: DIM C$(2,7)
110 INPUT "REBOBINE CINTA DE DATOS DESPUES
    DIGITE L",K$
120 IF K$<>"L" THEN GOTO 110
130 PROCLOAD
140 PRINT "TEMP DIA"
150 FOR F = 1 TO 7
160    FOR C = 1 TO 2
170       PRINT C$(C,F);"  "
180    NEXT C: PRINT
190 NEXT F
200 END
300 DEF PROCSAVE
310 X = OPENOUT ("DATTEMP")
320 FOR F = 1 TO 7
330    FOR C = 1 TO 2
340       PRINT # X,C$(C,F)
350    NEXT C
360 NEXT F
370 CLOSE # X
380 ENDPROC
400 DEF PROCLOAD
410 X = OPENIN ("DATTEMP")
420 FOR F = 1 TO 7
430    FOR C = 1 TO 2
440       INPUT # X,C$(C,F)
450    NEXT C
460 NEXT F
470 CLOSE #X
480 ENDPROC
```

Una de las ventajas del BASIC del BBC la constituye el hecho de que las sentencias para manipulación de archivos trabajan igualmente bien tanto para archivos en cassette como para archivos en disco.

Hasta el momento hemos visto cómo se pueden transferir datos de sentencias DATA, a través de matrices, a archivos de datos en cinta (o disco) y viceversa. El próximo paso será retomar el proceso de INICIALIZACION para ver exactamente cuántas matrices se necesitarán, cuántos elementos precisará cada una y en qué puntos del programa se habrán de transferir datos de y hacia ellas.

Ejercicio

Escriba usted un programa con los siguientes componentes:

```
EMPEZAR
   INICIALIZACION
   DAR ENTRADA A DATOS
   ELECCION
      Guardar datos
      Cargar datos
      Salir del programa
FIN
```

INICIALIZACION inicializará las matrices y variables que requiera el programa. Los datos comprenderán, digamos, 15 nombres, a los que se dará entrada desde el teclado mediante indicaciones en la pantalla. ELECCION le proporcionará al usuario un menú en el que le preguntará si desea ¿GUARDAR DATOS?, ¿CARGAR DATOS? o ¿SALIR DEL PROGRAMA? Vea si puede crear una "bandera" en la parte ¿SALIR DEL PROGRAMA? que guarde los datos si, y sólo si, aún no se hubiera efectuado una SAVE.

Complementos al BASIC

VARIABLES

Entre los ordenadores personales más populares, sólo el BBC Micro admite nombres de variables largos, como NOMBRE$. El Spectrum permite usar nombres largos para las variables numéricas, pero sólo nombres de una letra (además de "$") para las variables alfanuméricas. El Dragon 32, el Vic-20 y el Commodore 64 admiten nombres de variables largos, pero sólo son significativos los dos primeros caracteres, de modo que NOMBRE$ es válido, pero se refiere a la misma posición de memoria que NOMB$: las dos palabras tienen iguales los dos primeros caracteres.
En el Oric-1, los nombres de las variables no pueden estar compuestos por más de dos caracteres (primero una letra y luego un número o una letra). El Lynx admite sólo nombres de variables de una letra.

OPEN

CLOSE

PRINT#

En el Dragon 32 se debe utilizar este formato:

```
OPEN"0", # -1, "DATTEMP"
PRINT # -1,1,13.6,2,9.6,3,11.4, etc.
CLOSE # -1
```

y

```
OPEN"I", # -1, "DATTEMP"
INPUT # -1, D, T
CLOSE # -1
```

En el Commodore 64 y en el Vic-20 emplee este formato:

```
OPEN 1,1,2, "DATTEMP"
PRINT # 1,1,13.6,2,9.6,3,11.4, etc.
CLOSE 1
```

y

```
OPEN 1,1,0 "DATTEMP"
INPUT # 1, D, T
CLOSE 1
```

LYNX

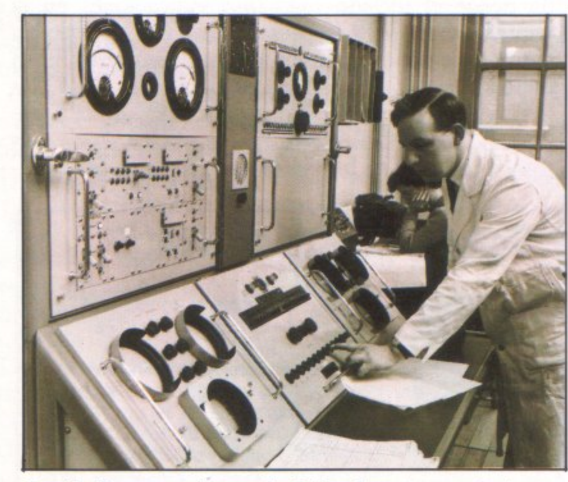
ORIC-1

El Lynx y el Oric-1, en su forma estándar, no admiten archivos en cassettes. No obstante, en el futuro saldrán al mercado ampliaciones que lo permitirán.

Tony Lodge

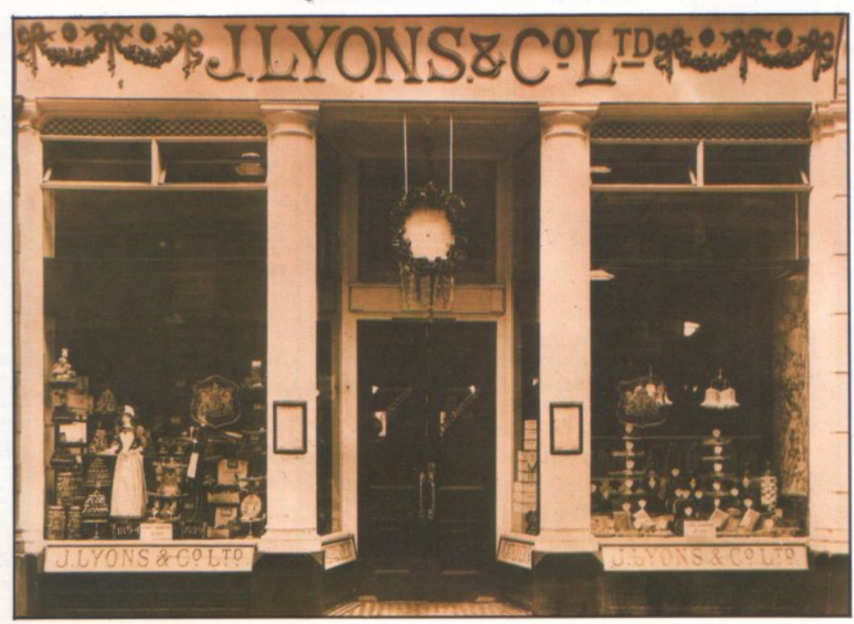
El LEO de Lyons

En Gran Bretaña, la informática comercial tuvo su origen... ¡en una tienda de comestibles!



Oficina electrónica
A diferencia de todos los ordenadores anteriores, que se habían diseñado para aplicaciones científicas o militares, el LEO 1 se creó para efectuar sólo operaciones aritméticas sencillas, pero sobre miles de ítems o transacciones por día

En 1947 se tomó una decisión futurista para intentar construir un ordenador que se pudiera utilizar para automatizar el trabajo de oficina de los dependientes. Sería el primer ordenador de gestión comercial del mundo. Esta imaginativa decisión se gestó en un lugar sorprendente: J. Lyons, una tienda de comestibles. El movimiento comercial de la cadena de tiendas Lyons comprendía una inmensa cantidad de pequeñas transacciones, y para que una organización empresarial de dichas características fuera rentable, era necesario mantener un estricto control sobre todo el trabajo administrativo. A título orientativo, consignemos el dato de que, incluso después de la devastación que para el Reino Unido supuso la segunda guerra mundial, la empresa contaba con más de 1 000 empleados encargados de clasificar los recibos de las tiendas.



Tienda de comestibles computerizada
La tradicional tienda de comestibles Lyons no parece un sitio muy apropiado para la primera gran aplicación comercial de la informática; pero fue precisamente este tipo de negocio, con su considerable número de pequeñas transacciones diarias, el que primero se interesó por los métodos informáticos

Lo cierto es que la firma Lyons tenía ya una larga tradición de inquietud innovadora en los métodos empresariales: ya en 1896 había introducido en sus establecimientos las máquinas de calcular y durante la década de los treinta experimentó con llevar registros de las transacciones en microfilm y fundó el primer centro de investigación empresarial para estudiar los métodos operativos.

Cada cierto tiempo, Lyons hacía viajar al extranjero a algunos representantes para que investigaran los nuevos adelantos que pudieran ser útiles para la empresa, y en 1947 envió a dos empleados a Estados Unidos para que se informaran acerca de los nuevos "cerebros electrónicos". El descubrimiento más útil que efectuaron fue que en su mismo país, en Cambridge, se estaba construyendo un ordenador.

La junta directiva de Lyons encargó un estudio de viabilidad para considerar el posible desarrollo de un ordenador propio para la empresa. El informe sugirió que se podía construir un ordenador por 100 000 libras y que éste supondría un ahorro de 50 000 al año. En consecuencia, en octubre de 1947 Lyons empezó a trabajar en el proyecto. El cometido era muy temerario, porque en aquel entonces el ordenador de Cambridge estaba sólo en la fase de diseño. Lyons le entregó a la Universidad de Cambridge una subvención de 3 000 libras para contribuir a la construcción de lo que se conocería como EDSAC (Electronic Delay Storage Automatic Computer). Dicho dinero se utilizó para adquirir del gobierno un excedente de válvulas. En 1949, el EDSAC concluyó con rotundo éxito su primera tarea: calcular una tabla de números primos.

Lyons analizó los problemas que habría de resolver su ordenador, esbozando las rutinas que serían necesarias. Estos estudios se convirtieron luego en los dibujos de ejecución de los primeros programas y ayudaron a determinar el diseño del hardware. Pero muy pronto se hizo evidente que un ordenador de gestión era fundamentalmente distinto de una máquina de investigación universitaria. Mientras que el EDSAC estaba diseñado para efectuar operaciones matemáticas largas y complicadas a partir de una entrada compuesta por unos pocos números, el ordenador de gestión debía resolver el tipo de problema opuesto. Las operaciones matemáticas eran mínimas (sumar y multiplicar), pero la cantidad de datos a procesar era enorme.

LEO (Lyons Electronic Office) entró en funcionamiento el 9 de febrero de 1954 y se utilizaba para calcular la nómina de pagos semanal de los 1 700 miembros del equipo de personal. Realizaba en un segundo y medio la misma operación que previamente efectuaba un empleado en ocho minutos.

Para Lyons, LEO representó un rotundo éxito y enseguida comprendieron que necesitaban más de una máquina. Otras firmas se mostraron interesadas y Lyons organizó una empresa que se dedicaría, aprovechando la experiencia adquirida, a fabricar y comercializar ordenadores. Leo Computers tuvo un gran éxito y amplió su campo produciendo una serie de versiones mejoradas de LEO que se utilizaron en la industria, en el mundo empresarial y en las oficinas públicas. En 1963 la empresa fue adquirida por la English Electric Company.

UNION PERFECTA

Así se comportan los periféricos creados por SINCLAIR para SINCLAIR: de forma perfecta. Y es lógico.

Cada vez que SINCLAIR diseña un microordenador, no lo hace de una manera aislada. Simultáneamente crea todos esos periféricos que van a hacer más potente, preciso y útil el microordenador que tiene entre manos.

Periféricos pensados y diseñados para dar un servicio óptimo, pero con un precio razonable, dentro de la filosofía SINCLAIR:

"Hacer la informática accesible a todos".

Por eso cuando creó el ZX 81 vio la necesidad de dotarlo con una ampliación de memoria de 16K RAM para que no quedara pequeño y de una impresora sencilla y barata pero útil y precisa.

Así es la filosofía SINCLAIR. Así son los periféricos de SINCLAIR para SINCLAIR.





CENTRAL COMERCIAL: Tomás Bretón, 60
Tel. 468 03 00 Telex: 23399 IYCO E Madrid.
DELEGACION CATALUÑA: Camp, 80 - Barcelona - 22

PARA JUGAR A LO GRANDE
(INSTANTANEAMENTE)

Presentamos **el Interface 2 ZX** Pensado y diseñado por SINCLAIR para unirse a la perfección con tu microordenador Spectrum.

Si a la hora de elegir tu microordenador optaste por el mejor, es lógico que elijas ahora el Interface 2 ZX

Ya habrás podido deleitarte con la más amplia variedad de juegos existentes para tu Spectrum (la más extensa del mercado). Ahora con el Interface 2 ZX vas a tener más ventajas para tu Spectrum:

- Podrás conectar Joysticks para sacarle, aún, mayor rendimiento a tus mejores juegos y divertirte con aquellos exclusivamente disponibles en **Cartuchos ZX:** correr, saltar, volar... a lo grande.
  ¡Menuda diferencia!
- Además, al ser cartuchos con memoria ROM, podrás, con tu SPECTRUM de 16 K, jugar con programas hasta ahora reservados para 48 K, sin ampliar la memoria.
  ¡Vaya ahorro!
- Al conectar el Interface 2 ZX tienes la certeza de poseer un periférico pensado por SINCLAIR para SINCLAIR. Tu microordenador queda a salvo de circuitos poco fiables.
  ¡Un alivio!
- Al adquirir el Interface 2 ZX y los Cartuchos ZX en la red de Concesionarios Autorizados, podrás exigir la tarjeta de garantía INVESTRONICA, única válida en territorio nacional.
  ¡Una tranquilidad!



Interface 2 ZX y Cartuchos ZX

Si aún no los tienes
no sabes lo que te pierdes

Solicita una demostración en cualquier Concesionario Autorizado INVESTRONICA.



CENTRAL COMERCIAL: Tomás Bretón, 60
Tel. 468 03 00 Telex: 23399 IYCO E Madrid.
DELEGACION CATALUÑA: Camp, 80 - Barcelona - 22